Improving OMR for Digital Music Libraries with Multiple Recognisers and Multiple Sources

Victor Padilla, Alan Marsden
LICA - Lancaster University
Lancaster Institute for the Contemporary Arts
Lancaster LA1 4YW, UK
+44 (0) 1524 510977/593774
v.padilla/a.marsden@lancaster.ac.uk

Alex McLean, Kia Ng
ICSRiM - University of Leeds,
Interdisciplinary Centre for Scientific Research in Music,
Leeds LS2 9JT, UK
+44 (0) 113 343 2572
a.mclean/k.c.ng@leeds.ac.uk

ABSTRACT

Large quantities of scanned music are now available in public digital music libraries. However, the information in such sources is represented as pixel data in images rather than symbolic information about the notes of a piece of music, and therefore it is opaque to musically meaningful computational processes (e.g., to search for a particular melodic pattern). Optical Music Recognition (Optical Character Recognition for music) holds out the prospect of a solution to this issue and allowing access to very large quantities of musical information in digital libraries. Despite the efforts made by the different commercial OMR developers to improve the accuracy of their systems, mistakes in the output are currently too frequent to make OMR a practical tool for bulk processing.

One possibility for improving the accuracy of OMR is to use multiple recognisers and combine the results to achieve an output better than each of them individually. The general process presented here can be divided into three subtasks, S1, S2, and S3. S1 is focused in the correction of rhythmical errors at bar level, counting the errors of the different OMR outputs, establish a ranking of the results, and make a pairwise alignment to select the best measures. S2 is based on the alignment and voting of individual symbols. For this task we have implemented a conversion of the most important symbols to a simple grammar. Finally, S3 improves the output of S2 by comparing and adding symbols from S1 and detecting gaps through the alignment of wrong measures.

The process described in this paper is part of our "Big Data Approach" where a large amount of data is available in music score libraries, such as the International Music Score Library Project (IMSLP), for the purpose of Music Information Retrieval (MIR).

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: performance evaluation (efficiency and effectiveness); I.7.5 **[Document and Text Processing]**: Document Capture—document analysis, graphics recognition and interpretation; J.5 **[Arts and Humanities]**: performing arts (music)

DLfM 2014, September 12th, 2014, London, UK.

General Terms

Algorithms, Performance, Reliability, Experimentation

Keywords

Optical Music Recognition, Pattern Recognition, Image processing, Library, Big Data

1. INTRODUCTION

Most current services and repositories which might qualify as Digital Music Libraries offer access to different digital representation of digital contents. Some focus on audio, for example, iTunes, Spotify and, in the classical domain, the Naxos Music Library. Some provide access to digitised/scanned image of scores, most notably the Petrucci Music Library (also known as the International Music Score Library Project, IMSLP[1]), and several public and academic music libraries, including the Sibley Music Library at Eastman School of Music, and University of Rochester.[2] Some services give access to metadata, such as the details of recordings and tracks available at MusicBrainz,[3] or the results of acoustic analysis at The Echo Nest.[4] The crucial information for musicologists, however, is often none of these three kinds, but the symbolic information of the scores such as the notes in the scores. A musicologist can read the symbolic information from the image of a score, or hear it from the audio recording, but if computational tools are to be used to search for or analyse music at this level of information, this kind of current digital music library is of little use.

There are collections which give access score symbolic information, such as MuseData, hosted at CCARH, Stanford University,[5] but they are small in number and limited in scope. None has shown the significant impact of community compilation which has made IMSLP so successful. The reason is probably because creating a digital file with such symbolic musical information manually is time consuming, requires specialist software and expertise, and generates significant new intellectual property.

The same issues apply in the domain of text, where collections may contain images of pages, recordings or speech, or metadata such as bibliographic details. In the case of collections of page

[1] www.imslp.org

[2] urresearch.rochester.edu/viewInstitutionalCollection.action?collectionId=25

[3] musicbrainz.org

[4] the.echonest.com

[5] www.musedata.org

images, though, Optical Character Recognition (OCR) technology is commonly applied to create searchable collections, and sometimes to generate symbolic files, e.g. Project Gutenberg.[6] Equivalent software for music, commonly called Optical Music Recognition (OMR) software, does exist, but its error-rate is typically too high for it to generate similar impact for digital music libraries, as OCR for the domain of text. (For an example of the potential impact of OMR, and an illustration of the limitations of current OMR software, see the Peachnote service for searching IMSLP.[7] [15])

The objective of our project, as introduced in [14], is to improve the quality of OMR by making use of multiple sources of information. We hope to do this by post-processing the results of multiple OMR software, and by adapting OMR software to make use of multiple inputs. In this paper, we describe our post-processing method, and report some preliminary results. It is unrealistic for us to hope that the project will improve OMR accuracy sufficiently to generate error-free large scale symbolic collections in digital music libraries, but we hope to make a contribution in this direction. At the very least, we hope to produce accuracy measures with realistic use-cases so that musicologists and collection curators can make informed decisions about the use of OMR.[8]

1.1 Previous OMR Research

OMR has been a topic of research for several decades. A recent survey and discussion of the problem can be found in [11]. Discussions on some of the challenges, mainly due to the complexity of scores in comparison with text and the combination between vertical and horizontal structures involved, can be found in Ng [10], Bellini [1], and Jones *et al.* [8] As noted above, the accuracy of OMR is limited, and even with a simple page as input, OMR software can produce errors. Those errors are difficult to predict and different in each case, even for the same page scanned at different resolutions.

Various post-processing approaches to improve the results have been proposed. Byrd & Schindele [2] suggest using rules for each OMR describing their strengths and weaknesses at different points such as grace notes, missing clefs, slurs, trills, and even establishing a ranking for a specific version of OMR. They propose a corpus of 17 rules, but recognise that 50 or even 100 rules would be a more reasonable number. A problematic aspect of this research is the moving target of new versions and the inaccuracy of old rules with respect to the new systems.

On the other hand, Bugge *et al.* [3] create a pipeline for converting the MusicXML output of each OMR to a new format, MusicXiMpLe (a subset of MusicXML keeping only the rhythm and pitch), and then converting them into symbolic sequences in a specific grammar for aligning and voting. They use the Needleman-Wunsch algorithm [11] extended to multiple sequences, and they report experiments with two different corpuses of music. One drawback of this system is that some information is lost in the format conversion pipeline. Proposed improvements include taking into account more parameters of the score, and to increase the grammar involved.

Obtaining information in the same score from different measures and parts was suggested by Church & Cuthbert [4]. The approach uses rhythmic repetition within a score to create a model where measure-level metrical errors can be fixed. They flag the incorrect measures and replace the errors with the most rhythmically similar material from another correct measure. It focuses on resolving one type of error at a time. For correcting pitch errors, another algorithm would be necessary. This system can be extended to multiple recognisers.

Ideas about the necessity of different approaches (*top-down* and *bottom-up*) in OMR are discussed in Jin *et al.* [7]. Rossant & Bloch [12] suggest a group of musical rules in fuzzy logic in order to obtain more consistent and reliable results.

2. GENERAL STRUCTURE

In this paper, we present one structure that combines the *bottom-up* and *top-down* classical approaches for solving problems. The general structure is divided into three subtasks (S1, S2 and S3), for clarity. S1 is top-down oriented, articulating and improving the output based on rules that identify the metrical correctness of the measure. The process or flagging wrong measures is not a straightforward decision due to elements such as anacrusis, repetition symbols in the middle of a measure or possible incomplete last measure in a movement. Besides, mistakes recognising time signatures in the middle of a score lead to an important number of "false positives" being flagged.

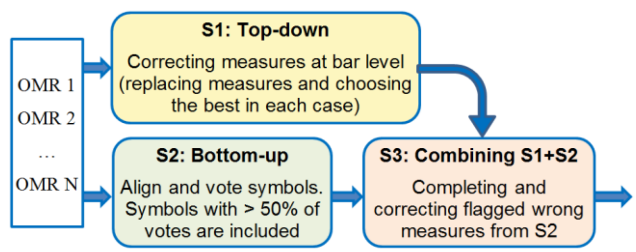

Figure 1. General flow structure.

S2 is a bottom-up strategy based on the idea of Bugge *et al.* [3] aligning and voting each symbol, without converting the musicXML output to different formats, and improving the process in some aspects like ties and tuplets. The main drawback of this approach is the impossibility of detecting incorrect decisions in the voting system (i.e., when two OMRs make the same error), and these errors will be propagated.

Most of the symbols voted in S2 tend to be correct, but important information is missing if the recognition rate is poor in some of the OMR. In the last step, S3, both strategies are combined completing the S2 measures with information from S1.

3. FLAGGING AND CORRECTING MEASURES. *TOP-DOWN (S1)*

Byrd & Schindele [2] comment on the problem of evaluating different OMR software and the relative importance of different kinds of error. At this point, our first approach is based on finding errors by checking for metrically incorrect measures. The most common errors are, for example, mistakes with dots, quavers recognised as semiquavers, missing notes, etc., and these errors

[6] www.gutenberg.org

[7] www.peachnote.com

[8] Software for automatic transcription from audio data rather than from images of pages exists for both text and music, and while both suffer from low accuracy, research in music transcription again lags behind the equivalent in text.

can be detected in most of the OMR recognisers as “measure errors”. In the particular case of PhotoScore, these errors are concealed in the MusicXML output through the adjustment of each measure with extra rests or removal and cropping of notes. (See Figure 2) For this particular OMR software, more information will be needed in order to detect and to flag the possible incorrect measures in the MusicXML output.

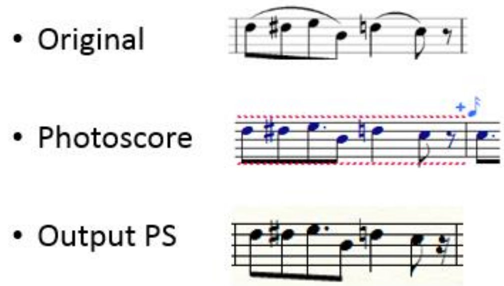

Figure 2. Example of PhotoScore recognition with rhythmical mistakes. The MusicXML output adjusts and corrects the measure cropping the last note.

In this research, we are testing four commercial OMR systems: Capella-Scan 8.0, PhotoScore Ultimate 7, SmartScore X2 Pro, SharpEye 2.68. In every case, our system processes the output of each of the above mentioned OMR system in MusicXML

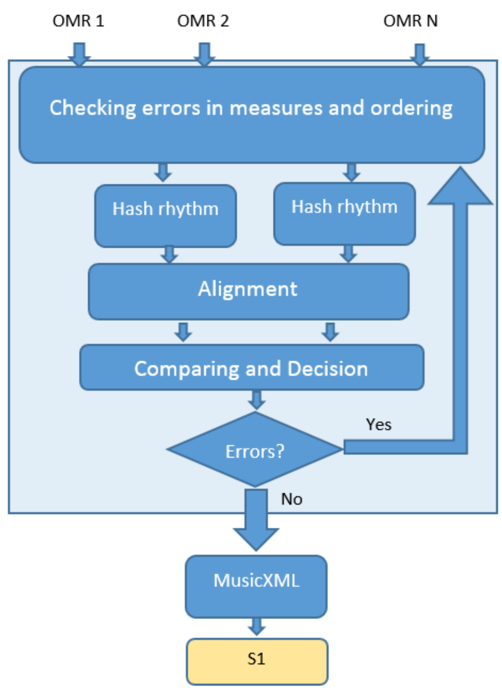

Figure 3. First part diagram for obtaining S1.

Once the number of incorrect measures is flagged in each OMR, a ranking is established. The best two OMRs are aligned and compared in order to obtain a new output (in MusicXML). If errors remain, the system continues comparing the output with the next OMR until there are no more errors, or the last OMR output has been compared (as illustrate in Figure 3).

3.1 Alignment of Measures

For each pair of OMR, it is necessary to align the measures in order to make proper comparisons. Many algorithms and tools for this general task have been developed, especially in association with biological sequences.[9] For this pairwise alignment, we use the classic Needleman-Wunsch algorithm, but any other algorithms, suitable to rhythmic sequences, can be used. For our first prototype, the sequences of each measure are converted to a hash of durations for simplicity in the pairwise-alignment and the pitches are removed (Figure 4). The hash sequence gives us rhythmic information, but it would be necessary to add pitch information for more general cases.[10] At the moment, we are testing with Mozart’s music, where the rhythmical patters are sufficiently varied to govern good results.

['Q', 'd', '`P', 'PQZ', 'VFFFFF', '`P']

Figure 4. Sequence of measures converted to hash array.

The Needleman-Wunsch algorithm is defined as

$$M_{i,j} = \max \left| \begin{array}{c} M_{i-1,j-1} + \alpha(seq1[i], seq2[j]) \\ M_{i-1,j} + g \\ M_{i,j-1} + g \end{array} \right. \qquad (1)$$

M is the similarity matrix of the measures, *i*, *j* are the measure numbers and *g* is the gap penalty, in this case $g = -1$

$$-1 < \alpha < +1 \qquad (2)$$

where **α** is a measure of similarity between bars and is calculated in this case using the Needleman-Wunsch algorithm. Figure 5 shows an example of alignment with two OMR outputs to determine the value of **α**.

		(OMR 1 Bar 4)			
			P	Q	Z
(OMR 2 Bar 3)		0	-1	-2	-3
	Q	-1	-2	0	-1
	Z	-2	-3	-1	1

Figure 5. Example alignment of sequences at bar level.

[9] Clustal is a widely used multiple sequences alignment program for biological research, http://www.clustal.org/. Another program is mafft (http://mafft.cbrc.jp/alignment/software/) that allows working with any multiple sequences.

[10] This alignment through rhythmic hash sequences would not be appropriate for rhythmically even music, for example, a long sequence of semiquavers such as in the Prelude of Bach’s suite nº1 for solo Cello.

The measure of the similarity $\boldsymbol{\alpha}$ can be obtained from (3) below:

$$\alpha = M_{i,j}/l \qquad (3)$$

where l is the length of the shorter sequence. In this example, the result is $\alpha = 0.5$. If $\alpha < -1$ we set the value to -1, in which case, we understand that both sequences are completely different. This simple algorithm provides acceptable results in the alignment of rhythmic hash sequences. Other algorithms based on distance will be tested in the future.

		(OMR 1)	Bar 1	Bar 2	Bar 3	Bar 4	Bar 5
			'Q'	'd'	'`P'	'PQZ'	'VFFF
(OMR 2)		0	-1	-2	-3	-4	-5
Bar 1	'Q'	-1	1	0	-1	-2	-3
Bar 2	'd'	-2	0	2	1	0	-1
Bar 3	'QZ'	-3	-1	1	0	1.5	0.5
Bar 4	VFFF	-4	-2	0	0	0.5	2.5

Figure 6. Example of comparison and alignment two hash sequences of different OMR.

After the matrix is filled with the values, the alignment is determined by tracing a path from the bottom-right cell back to the top-left, selecting the highest-value neighbouring cell in each case. The value of the bottom-right cell provides an idea of similarity between the two sequences. With the same equation (3) the system obtains a value that indicates the similarity between the two OMR outputs. In the example as shown in Figure 6, this is $2.5/4 = 0.625$.

Once the two OMR outputs have been aligned, the next step is to take the better measures and to remove extra bars (gap penalties). Continuing with our example, it can be observed how the OMR 2 has introduced an extra measure at bar 3, due to the alignment algorithm (see Figure 7). Probably, the OMR 1 has made a mistake recognising a crotchet as a bar line. In order not to propagate errors, all these extra measures are removed for the next comparison.

The final result is a new MusicXML output that will be compared with the next OMR output, if the general measure errors are higher than zero, and this is repeated in a recursive process.

	Bar 1	Bar 2	Bar 3	Bar 4	Bar 5
(OMR1)	Q	d	P	PQZ	VFFF
(OMR 2)	Q	d	----	QZ	VFFF

Figure 7. Example alignment with extra measure in OMR 2 at bar 3.

4. ALIGNING AND VOTING. *BOTTOM-UP (S2)*

This second building block is based on the idea of removing errors based on common symbols. For the alignment of sequences, the same Needleman-Wunsch algorithm described before is applied, but extending the pairwise alignment. If 50% or more of OMR agree, the symbol is flagged as correct. In the particular case of having 4 OMR systems, the decision between >50% or >=50% is not trivial and produce completely different results. If three of them have to be in agreement, the recognition rate is lower, but more accurate (the symbols included have higher probabilities of being correct). If only two have to agree, the output produces more symbols, but propagates more errors.

In order to align and vote symbols easily, we have created a simple grammar of the most important elements in the score. Expressions and slurs are removed, but ties and tuplets remain even though they are not taken into account in the alignment process. If the symbol voted has secondary information (ties, tuplets or elements of repetition bars), this information is maintained. The syntax used is very simple for identifying each symbol. The structure is as follows:

[[symbolType:basicInform.][extra 1][extra 2]…[extra n]] (4)

where

$$symbolType \in \{TS, KS, N, R, CH, CL\}$$

and *basicInform* is the basic information of the symbol. In the case of notes:

[N: *pitch_duration*]

where the duration is specified in quarter length (crotchet=1, quaver=0.5, semiquaver=0.25, …). Dots are included in this duration, but triplets are not, Instead, they are included in the second position of the array. *Extra 1, 2…n* provides additional information important for each symbol, but not essential in the alignment.

Some examples:

- Time signature: [TS:3/4] (3/4 time signature)
- Key signature: [KS:2] (2 sharps in the key signature)
- Notes: [[N:G5_0.25][0.166][‘start’]]
  (G5 triplet semiquaver with tie)
- Rests: [[R:0.25][0.166]]
  (Triplet semiquaver rest)
- Bars: [[‘!’][‘repeat’][‘start’]]
  (double bar start repetition)

For the alignment algorithm, only the first value in the array is taken. In the case of notes, the second value indicates the real duration (tuplets) and the third ties. The same idea is implemented in rests, bars and chords. Once the symbols are aligned, the ones with the most votes (50% or more) are included in the output and converted back to MusicXML.

5. COMBINING *(S3)*

The S2 strategy (aligning and voting) guarantees that most of the symbols included are correct, but many of them can be missing. At this point, we are not correcting symbols in S2, and we are taking this output as an incomplete truth. With a simple “alignment and complete” algorithm between S1 and S2, some of them can be recovered. For the alignment of the measures, the algorithm presented in 3.1 has been implemented. The alignment inside each measure is made by transforming the measures to sequences of symbols, as was explained in section 4, and aligning them using the Needleman-Wunsch algorithm. In the alignment, the gaps produced in the measures of S2 are filled with the information of S1. A more advanced algorithm will be implemented for detecting missing bars because, in this case, it is necessary to determine the point for splitting the measure.

Figure 6 illustrates the combination of outputs from S1 and S2. The red symbols are incorrect, and the green ones are missing. In this particular case, the second measure of S1 is completely wrong (missing) and the anacrusis is misplaced, but the rest of the notes are correctly recognised. Aligning and comparing symbols, the sequence of the second measure will be:

S1: [!][N:A4_1.0][N:D5_1.0][N:D5_2.0][!]

S2: [!][N:A4_1.0][N:D5_1.0][*][!]

Replacing *[*]* in S2 by the appropriate symbol from S1 *[N:D5_2.0]* the mistake is corrected.

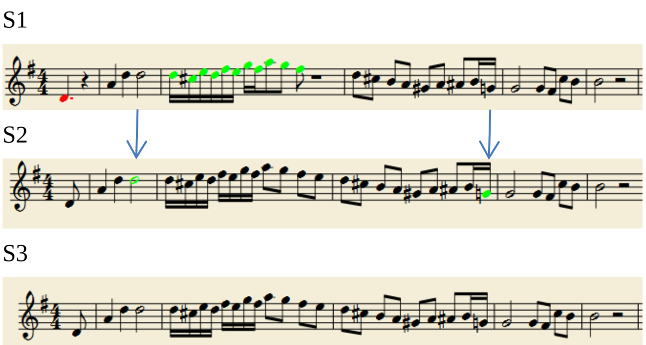

Figure 6. Example of combination S1+S2 to improve the output.

5.1 Rules for detecting wrong measures

A better degree of improvement is achieved by identifying correctly as many wrong measures possible, but this is not always a trivial task. The detection of incorrect measures was made in the first approach using the *correctors* class of music21, but there are many cases where flagging wrong measures based on the simple time signature mark of the score produces bad results.

Some algorithms are proposed here.

a) Anacrusis and repetition marks

Anacrusis or repetition marks in the middle of the measure are easy to detect and implement if the general time signature is correctly recognised. The simple algorithm of adding two adjacent measures and checking if the value is equals to a complete measure, works appropriately in most of the cases

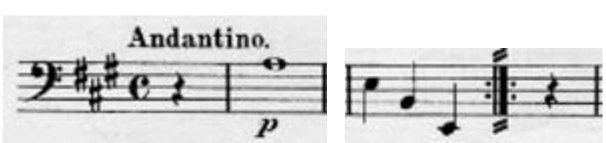

Figure 7. Anacrusis and repetition marks detected as "false positive" errors.

b) Finding the appropriate time signature

There are many cases where the time signature information is lost or incorrect. To detect this case, an algorithm based on the average measure value is required. As a first approach, the algorithm detects and flags possible transitions based on the average number of quavers. If two transitions are too close, less than an estimated value, one of them is removed. These transitions allow the stream of measures to be divided into chunks. Finally, the system calculates the average number of quavers of each chunk and estimates the time signature to be compared with each measure.

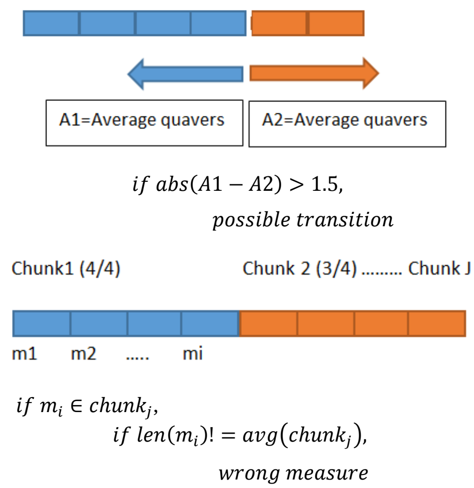

Figure 8. Algorithm for detecting wrong measures based on the context.

c) Implementing stylistic rules

Another type of rule can be implemented based on the musical style. Figure 8 shows an example of a measure with a typical sequence of semiquavers in a classical imitative progression. In the output of the OMR, one semiquaver is wrongly converted to a quaver and the last note is removed. Three things point to potential mistake(s) here: the rhythmic pattern is very unusual, the beaming does not follow the usual rules, and while the sequence of pitches in the second half of the measure clearly follows the pattern in the first half, the rhythm does not. While this is not flagged as an error by the OMR software (because the measure fits the expected duration in 4/4) the mistake could be easily detected and corrected by reference to a library of possible musical stylistic rules.

Figure 8. Rhythmic error in a 4/4 measure.

d) Extracting the appropriate information

A more complex situation is when the tuplets are implicit but not explicit (Figure 9). If one OMR recognises perfectly all the symbols a simple algorithm for detecting wrong measures would indicate a “false positive”. Furthermore, things become more complex when some OMRs try to guess the real rhythm by adjusting the 5 semiquavers, making them a quintuplet or introducing a new time signature (Figure 9). A future system for detecting implicit rhythms, possibly based on the beams and the style, will be implemented.

Figure 9. Triplets implicit. Photoscore changes the time signature from 4/4 to 9/8.

6. TESTING THE SYSTEM

One of the most difficult, and controversial, parts of research about OMR is how to define and measure improvement in a new system. Typical mistakes such as extra bars, missing notes or extra dots cannot be equally weighted, and even the position of the error in the bar should affect the final mark. For this purpose, we have developed an application for counting missing and wrong symbols equally and identify them in colours (red and green) in the XML output file. The system aligns and compares the output with the ground truth and determines the percentage of accuracy.

The testing process can be divided into the following steps:

a) Creating the MusicXML ground truth

The first problem is the availability and reliability of sources to create a sufficiently large set of ground truth to test the system. In many cases, the sources are in musedata format [6] with basic symbols (rhythm and pitch without slurs, ties, expression markings, etc…), and mistakes can be generated in the automatic transcription to MusicXML. In fact, in our experience correcting by hand the Mozart string quartets K. 387 and K. 421 from the music21 corpus in musedata format, the conversion to MusicXML makes some mistakes in the rhythm, especially with some dotted half notes.

Another important point is the existence of different versions or editions of a piece. Even assuming that the pitches and rhythms are the same,[11] rest symbols, dots and ties can be slightly different, although equivalent. Extra adjustments will be made to prevent “false positives” in the testing process. For the initial tests reported here, we have used two editions of Mozart’s string quartet Nº 14 in G major, K. 387 and Peters edition of K. 421 from IMSLP. The ground truth was created by hand using music-notation software to generate MusicXML.

b) Calculating the differences between the ground truth and each OMR

Each OMR is aligned with the ground truth, and the system calculates the mistakes producing a number that reflects the similarity. As noted above, not all errors in OMR have the same importance and a more detailed process will be evaluated in the future in order to weight them appropriately.

The symbols included are notes, chords, rests, barlines, key signatures and time signatures. In this first assessment tuplets and ties are not evaluated.

c) Calculating the differences between the ground truth and our multiple OMR

This procedure is similar to step b. The number obtained indicates the accuracy and allows automatic comparison with other scores and other OMR. From this comparison, the relative percentage of improvement is obtained without checking or counting errors by hand. This will be essential in a later stage of our research in which we aim to test our methods on large quantities of data.

6.1 Results

We have tested our system with 64 pages of music using the Mozart’s string quartet Nº 14 in G major, K.387 and Nº 15 in D minor, K. 421. In this first version, we focused on parts rather than full scores for ease of checking the errors. The OMR programs involved in this case are the four mentioned in Section 3: Capella, SharpEye, SmartScore and PhotoScore.

Figures 8 and 9 show the results. The score sources used were PDF downloaded from IMSLP and converted to raster images. The first column group in Figure 9 shows results of a PDF produced by an IMSLP user ‘Gory’ using the Finale typesetting software. It is very clear and clean, with a resolution of 150 dpi. The other column groups are for scans of the Peters edition from c. 1882, at a resolution of 300 dpi. It is also relatively clean but the edges of the line symbols are jagged, the layout is more compressed, and there are symbols that are not present in the first source, such as fingering and bowing.

In many cases, the resolution of the image can decisively affect the accuracy of the recogniser, but there are no clear relations. Sometimes higher resolutions are better and sometimes worse. It could be possible to find the best recognition rate, for each OMR and page, using a trial-and-error system through an algorithm that iterates between several resolutions and obtains the best results. This iterative system could be another idea to be implemented in the future.

These results evaluate only the S2 output. The average of recognition is around 93.4% versus 89.5% for the best OMR. The improvement rate is around 4 percent which means that in an average page with 500 symbols, 20 have been corrected. The S1-S3 system is still in the process of further tuning and implementation of appropriate rules. Our first estimates indicate that an improvement of around 1% or 2% in S1-S3 over S2 can be achieved, to correct another 4 to 7 symbols/page, but it is possible that we are approaching maximal possible recovery of information via this kind of technique. In our tests using scores from the Peters edition, the average OMR recognition rate is lower than 88%. This excludes results from Capella on the viola parts because the clefs signs have not been correctly recognised.

[11] This is not the case for many scores. As an example, the BWV 853 – Well-Tempered Clavier, Book 1: Prelude and Fugue No. 8 in E-flat minor can be found in D-sharp minor in many editions.

Table1. Recognition rate in different OMR and S2 output

	CP	PS	SE	SS	S2
Gory (15p)	89.67	95.02	90.38	90.55	95.33
k387_cello (6p)	83.44	89.61	88.07	90.54	93.25
k387_v I (9p)	82.65	88.62	85.52	87.27	92.19
k387_v II (6p)	85.45	89.54	87.08	89.35	94.07
k387_vla (6p)		87.41	88.38	86.99	89.84
k421_cello (4p)	85.96	90.23	90.37	91.06	96.31
k421_v I (6p)	80.43	87.24	85.46	89.14	93.83
k421_v II (6p)	80.67	88.24	85.35	90.28	93.07
k421_vla (6p)		89.99	89.09	89.82	92.74
Average	84.04	89.54	87.74	89.44	93.40

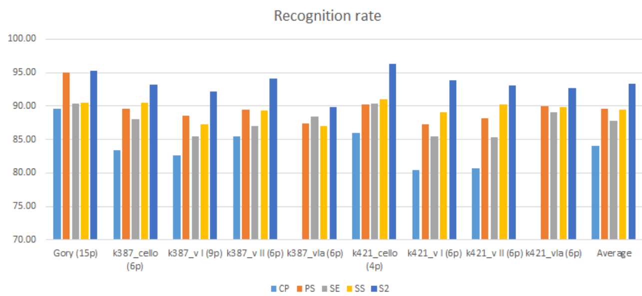

Figure 9. Recognition rate from different OMR and S2 output.

7. NEXT STEPS

The system implemented is focused on metrical errors at measure level. The source code of this version can be downloaded from http://github.com/MultiOMR, but this is only a "work in progress". There are considerable works still to be done in order to achieve a sufficiently accurate system with a reliable procedure, that can processes a large amount of images available in digital musical libraries in batch without manual interventions.

This project is part of our Big Data approach that explores two directions to improving OMR accuracy. The research reported here is the first direction, utilising post-processing of the outputs of multiple OMR programs and multiple image sources to improve the accuracy of the final output. The second approach will combine information from several source images as input to the recogniser, so as to increase the accuracy of the recogniser itself. At this early point in the research, many difficulties have to be addressed. We outline some of these below.

7.1 Towards an Automatic and Extended Process

The majority of OMR tools are proprietary, which places limits on the extent to which we are able to extend and modify them. This presents challenges, not only for creating any automatic "batch process", but also in the correct transcription of the output. Nonetheless, a key requirement is for us to automate the process of indexing and accessing the images available online, where necessary pre-processing them into the input format required by each OMR software (e.g. SharpEye requires tiff input as separate pages), and collecting and transforming the output into a form suitable for our multiple OMR is not a straightforward task.

Of particular concern is the amount of data we have to process. The overall system will naturally be modular, with various stages of processing in a pipeline, potentially running across different operating systems. The systems architecture for this system will be based on industry standard free/open source systems, such as the cross platform ZeroMQ for high performance, asynchronous messaging between modules, the Postgres traditional SQL database, and potentially the Mogile distributed filesystem.

7.2 Optimising the Algorithm Performance

The structure implemented is separated in three different sub-processes, but redundant tasks are involved. For a prototype, it is interesting to have three processes separated in order to analyse results. However, for a larger scale, good performance is a crucial point. In the next implementation, the subtasks S1 and S3 will be merged and applied only to wrong measures from S2.

7.3 Pre-processing of Score Images based on Multiple Sources

Part of our team is working in the improvement of images based on different sources. In fact, this second research will be the first building block of our project. The pre-processing of images will provide an improved input to the multiple OMR system. The general idea is to perform a prior segmentation and analysis of images in a fast and reliable fashion, and use this information to inform later decisions in the multi-OMR process.

For printed scores, we can generally assume roughly parallel horizontal lines constituting staffs, with vertical lines that cut though one or more staffs to form a block structure that can be detected and segmented from one another. Through application of image processing techniques, we will be able to segment images into their constituent bars during such a pre-processing stage and feed them into multi-OMR separately. We will then be certain that the outputs from multiple OMR are from the same part of the score, without relying upon a post-alignment step.

Having broken a score down into measure segments, we will be able to apply further image processing techniques, applying metrics to profile each segment. We expect to find several uses for such data, with focus on matching bars across several scores of the same music, so that multiple scores can be fed into OMR, and the results merged towards higher accuracy. We will evaluate metrics which are stable across multiple scans of the same score. Our hypothesis is that naïve approaches to image processing, such as image moment and axis of orientation, will prove robust despite taking little account of musical content of the images.

7.4 Profiling OMR Tools

By building a profile of the strengths and weaknesses of each OMR tool, we will be able to calibrate multi-OMR accordingly. For example, we can rank the performance of OMR tools for dealing with poor quality scores, perhaps by introducing additional noise to source images, and comparing the degradation in the quality of OMR results from each tool. We would then be able to analyse actual source images to measure aspects of scan quality, and take this into account when merging the resulting OMR output.

A crucial part of our project will be to automate the process of profiling, using on-line sources of image and symbolic data, with appropriate estimates of the accuracy of the symbolic data as ground truth. This will allow us to produce more reliable and comprehensive data on the accuracy of different OMR tools than has been possible in previous smaller scale research (e.g., [2] and [9].)

8. CONCLUSIONS

OMR is the only realistic means of making the information in the vast quantity of music scores held by libraries open to digital processing. However, as indicated above, the current state of the art does not produce sufficiently accurate results to make it worthwhile for any library to undertake a significant project to convert its holdings to digital symbolic form.

The research reported here is part of a project which takes a different approach to traditional OMR which attempts to recognise symbolic information from an image of a score. Instead we focus on exploiting multiple sources of information. Around 35% of the pieces of music in IMSLP have some kind of multiple information, e.g. scans of different editions, more than one scan of the same score, scans of full score and parts, and/or other combinations. This redundant information, in combination with the outputs of different OMR software, multiplies the amount of data to process, but improves the possibility of producing an accurate result.

The experiments reported here are preliminary and in a small scale, but the results are encouraging. While we do not envisage that this project alone will produce a tool sufficiently accurate for general use in digital music libraries, we do hope to make significant progress in that direction, and we do hope to produce software which will make it easier for the dedicated musicologist to compile symbolic datasets for computational music research.

9. ACKNOWLEDGEMENTS

This work is conducted with funding support from the Arts and Humanities Research Council (AHRC; AH/L009870/1). Thanks to support from the following OMR systems: the Optical Music easy Reader (OMeR) and PDFtoMusic, for a copy of their software.

10. REFERENCES

[1] Bellini, P., Bruno, I., and Nesi, P. 2007. Assessing optical music recognition tools. *Computer Music Journal*, *31*(1), 68-93.

[2] Byrd, D., & Schindele, M. 2006. Prospects for Improving OMR with Multiple Recognisers. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006)*, Victoria, Canada, 41–46. Revised and expanded version (2007) retrieved February 20, 2013, http://www.informatics.indiana.edu/donbyrd/MROMRPap.

[3] Bugge, E. P., Juncher, K. L., Mathiasen, B. S., and Simonsen, J. G. 2011. Using sequence alignment and voting to improve optical music recognition from multiple recognisers. In *Proceedings of the 12th International Society for Music Information Retrieval* (Miami, FL, USA, October 2011), 405–410.

[4] Church, M., and Cuthbert, M. 2014. Improving Rhythmic transcriptions via probability models applied Post-OMR. Forthcoming in *Proceedings of the 15th International Society for Music Information Retrieval* (Taipei, Taiwan, October 2014)..

[5] Cuthbert, M., and Ariza, C. 2010. music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data. In *Proceedings of the 11th International Society for Music Information Retrieval Conference* (Utrecht, Netherlands, August 2010) 637–42.

[6] Hewlett, W., *et al*. 1994, 2000. MuseData: an Electronic Library of Classical Music Scores, http://musedata.org.

[7] Jin, R., and Raphael, C. 2012. Interpreting rhythm in optical music recognition. In *Proceedings of the 13th International Society for Music Information Retrieval Conference*

[8] Jones, G., Ong, B., Bruno, I., and Ng, K. 2008. Optical Music Imaging: Music Document Digitisation, Recognition, Evaluation, and Restoration. In *Interactive Multimedia Music Technologies*, K. Ng & P. Nesi Eds. Hershey, PA: Information Science Reference, 50-79. DOI= http://dx.doi.org/10.4018/978-1-59904-150-6.ch004.

[9] Knopke, I., and Byrd, D. 2007. Towards mucisdiff: a foundation for improved optical music recognition using multiple recognisers. In *Proceedings of the 8th International Conference on Music Information Retrieval* (Vienna, Austria, September 2007) 122–126.

[10] Ng, K. 2004. Optical music analysis for printed music score and handwritten music manuscript. In *Visual Perception of Music Notation: On-Line and Off-Line Recognition*, S. E. George, Ed. IGI Global, Hershey, PA, 108-127. DOI= http://dx.doi.org/10.4018/978-1-59140-298-5.

[11] Rebelo, A., Fujinaga, I., Paszkiewicz, F., Marcal, A.R.S., Guedes, C., and Cardoso, J.S. 2012. Optical music recognition: state-of-the-art and open issues. *Int. J. Multimed. Info. Retr*.1, 3 (October, 2012), 173-190. DOI= http://dx.doi.org/10.1007/s13735-012-0004-6.

[12] Rossant, F., and Bloch, I. 2007. Robust and adaptative OMR system including fuzzy modeling, fusion of musical rules and possible error detection. *EURASIP Journal on Advances in Signal Processing*

[13] Needleman, S. B., and Wunsch, C. D. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48, 3, 443–453, March.

[14] Ng, K., McLean, A. and Marsden, A. 2014. Big Data Optical Music Recognition with Multi Images and Multi Recognisers. In *Proceedings of Electronic Visualisation and the Arts* (London, UK, 2014), 215-218. DOI=http://dx.doi.org/10.14236/ewic/eva2014.26

[15] Viro, V. 2011. Peachnote: Music score search and analysis platform. In *Proceedings of the 12th International Society for Music Information Retrieval Conference* (Miami, FL, USA, October 2011), 359-362.