IMPROVING OPTICAL MUSIC RECOGNITION BY COMBINING OUTPUTS FROM MULTIPLE SOURCES

Victor Padilla
Lancaster University
victor.padilla.mc@gmail.com

Alex McLean
University of Leeds
a.mclean@leeds.ac.uk

Alan Marsden
Lancaster University
a.marsden@lancaster.ac.uk

Kia Ng
University of Leeds
k.c.ng@leeds.ac.uk

ABSTRACT

Current software for Optical Music Recognition (OMR) produces outputs with too many errors that render it an unrealistic option for the production of a large corpus of symbolic music files. In this paper, we propose a system which applies image pre-processing techniques to scans of scores and combines the outputs of different commercial OMR programs when applied to images of different scores of the same piece of music. As a result of this procedure, the combined output has around 50% fewer errors when compared to the output of any one OMR program. Image pre-processing splits scores into separate movements and sections and removes ossia staves which confuse OMR software. Post-processing aligns the outputs from different OMR programs and from different sources, rejecting outputs with the most errors and using majority voting to determine the likely correct details. Our software produces output in MusicXML, concentrating on accurate pitch and rhythm and ignoring grace notes. Results of tests on the six string quartets by Mozart dedicated to Joseph Haydn and the first six piano sonatas by Mozart are presented, showing an average recognition rate of around 95%.

1. INTRODUCTION

Musical research increasingly depends on large quantities of data amenable to computational processing. In comparison to audio and images, the quantities of symbolic data that are easily available are relatively small. Millions of audio recordings are available from various sources (often at a price) and images of tens of thousands of scores are freely available (subject to differences in copyright laws) in the on-line Petrucci Music Library (also known as IMSLP). In the case of data in formats such as MEI, MusicXML, Lilypond, Humdrum kern, Musedata, and even MIDI, which give explicit information about the notes that make up a piece of music, the available quantities are relatively small. The KernScores archive [21] claims to contain 108,703 files, but many of these are not complete pieces of music. Mutopia, an archive of scores in the Lilypond format, claims to contain 1904 pieces though some of these also are not full pieces. The Musescore collection of scores in MusicXML gives no figures of its contents, but it is not clearly organised and cursory browsing shows that a significant proportion of the material is not useful for musical scholarship. MIDI data is available in larger quantities but usually of uncertain provenance and reliability.

The creation of accurate files in symbolic formats such as MusicXML [11] is time-consuming (though we have not been able to find any firm data on how time-consuming). One potential solution to this is to use Optical Music Recognition (OMR) software to generate symbolic data such as MusicXML from score images. Indeed, Sapp [21] reports that this technique was used to generate some of the data in the KernScores dataset, and others have also reported on the use of OMR in generating large datasets [2, 3, 6, 23]. However, the error rate in OMR is still too high. Although for some MIR tasks the error rate may be sufficiently low to produce usable data [2, 3, 23], the degree of accuracy is unreliable.

This paper reports the results of a project to investigate improving OMR by (i) image pre-processing of scanned scores, and (ii) using multiple sources of information. We use both multiple recognisers (i.e., different OMR programs) and multiple scores of the same piece of music. Preliminary results from an earlier stage of this project were reported in [16]. Since then we have added image pre-processing steps, further developments of the output-combination processes, and mechanisms for handling piano and multi-part music. We also report here the results of much more extensive testing. The basic idea of combining output from different OMR programs has been proposed before [4, 5, 13] but this paper presents the first extensive testing of the idea, and adds to that the combination of outputs from different sources for the same piece of music (different editions, parts and scores, etc.).

In view of our objective of facilitating the production of large collections of symbolic music data, our system batch processes the inputted scores without intervention from the user. The basic workflow is illustrated in Figure 1. Each step of the process is described in subsequent sections of this paper, followed by the results of a study that tested the accuracy of the process.

Music notation contains many different kinds of information, ranging from tempo indications to expression markings, and to individual notes. The representation varies in both score and symbolic music data formats. In this study we assume the most important information in a score to be the pitch and duration of the notes. Therefore, we have concentrated on improving the accuracy of recognition of these features alone. Grace notes, dynamics, articulation, the arrangement of notes in voices, and other expression markings, are all ignored. However, when a piece of music has distinct parts for different instruments (e.g., a piece of chamber music) we do pay attention to those distinct parts. For our purposes, a piece of music therefore consists of a collection of "parts", each of which is a "bag of notes", with each note having a "pitch", "onset time" and "duration".

Broadly speaking, we have been able to halve the number of errors made by OMR software in recognition of pitches and rhythms. However, the error rate remains relatively high, and is strongly dependent on the nature of the music being recognised and the features of the inputted score image. We have not tested how much time is required to manually correct the remaining errors.

2. BACKGROUND

2.1 Available Score Images

In the past it was common for research projects to scan scores directly. For example, in [2, 3], pages were scanned from the well-known jazz 'Fake Book'. Now, however, many collections of scans are available on-line. The largest collection is the Petrucci Music Library (also called IMSLP),[1] which in April 2015 claimed to contain 313,229 scores of 92,019 works. Some libraries are placing scans of some of their collections on-line and some scholarly editions, such as the Neue Mozart Ausgabe (NMA),[2] are also available on-line. Scores available online are usually of music which is no longer in copyright, and date from before the early twentieth century.

Most of these scans are in PDF format and many are in binary images (one-bit pixels). Resolution and qualities of the scans varies.

2.2 OMR Software

Eight systems are listed in a recent survey as 'the most relevant OMR software and programs' [18]. Of these we found four to be usable for our purpose: Capella-Scan 8.0, SharpEye 2.68, SmartScore X2 Pro and PhotoScore Ultimate 7.[3] All four pieces of software produce output in MusicXML format [11]. They differ in the image formats which they take as input, and also in whether they can take multiple pages as input. The lowest common denominator for input is single-page images in TIFF format.

Although our objective was not to evaluate the different OMR programs, we did find that the programs differed considerably in their accuracy when applied to different music. No one OMR program was consistently better than the rest. An indication of the differences between them is given in the results section below.

3. OUR MULTIPLE-OMR SYSTEM FOR IMPROVED ACCURACY

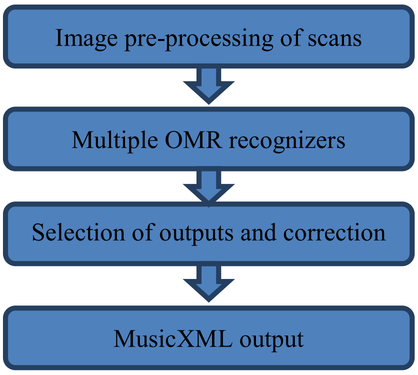
Image pre-processing of scans
Multiple OMR recognizers
Selection of outputs and correction
MusicXML output

Figure 1. Basic workflow of the proposed system.

3.1 Image Pre-Processing

As stated above, the common required input for the OMR programs is single-page TIFF images. The first steps in the image pre-processing are therefore to split multiple-page scores into individual pages, and to convert from PDF, which is the most common format used for downloadable score images, including those from IMSLP.

The other pre-processing steps depend on the detection of staves in the image. In general we use the Miyao [14] staff finding method as implemented in the Gamera software,[4] which locates equally spaced candidate points and links them using dynamic programming. This method did not perform well at detecting short ossia staves. For this we applied the Dalitz method (in class StaffFinder_dalitz) from the same library [9].[5]

Further processing is required to recognise systems in the images. Contours are detected using the findContours function [22] of the OpenCV library for computer vision,[6] with those containing staves marked as systems. Each of the remaining contours are then assigned to the nearest system, looking for the largest bounding box overlap, or simply the nearest system on the y-axis.

None of the OMR programs used handled divisions between movements properly: the music in an image was always assumed by the software to be a single continuous

[1] www.imslp.org

[2] dme.mozarteum.at

[3] www.capella.de, www.visiv.co.uk, www.musitek.com, www.sibelius.com/products/photoscore The other four listed by Rebelo et al. are ScoreMaker (cmusic.kawai.jp/products/sm), which we found to be available only in Japanese, Vivaldi Scan, which appears to have been withdrawn from sale, Audiveris (audiveris.kenai.com), open-source software which we found to be insufficiently robust (version 5 was under development at the time of this project), and Gamera (gamera.informatik.hsnr.de), which is not actually an OMR system but instead a toolkit for image processing and recognition.

[4] gamera.informatik.hsnr.de

[5] music-staves.sf.net

[6] opencv.org

piece of music. This is not problematic in a process which depends on the intervention of a human operator. However, this is inefficient and not scalable for large-scale batch processing. Therefore, we implemented a process which recognises the beginnings of movements, or new sections, from the indentation of the first system of staves. Where indented staves are detected, the output is two or more TIFF files containing images of those staves which belong to the same movement or section. This procedure correctly separated all cases in our test dataset.

A second common source of error was found to be 'ossia' segments. An ossia is a small staff in a score, generally placed above the main staff, that offers an alternative way of playing a segment of music, for example, giving a possible way of realising ornaments. The OMR programs tended to treat these as regular staves, leading to significant propagation errors. Since, as indicated above, our aim was to improve the recognition accuracy of pitches and rhythms only, ossia staves would not contain useful information consistent with this aim. The best course of action was to simply remove them from the images. Therefore, the minimum bounding rectangle which included any staff that was both shorter than the main staff and smaller in vertical size, and any symbols attached to that staff (in the sense of there being some line of black pixels connected to that staff), was removed from the image. We did not separately test the ossia-removal step, but found that a few cases were not removed properly.

3.2 Post-Processing: Comparison and Selection

The images resulting from the pre-processing steps described above are then given as input to each of the four OMR programs. Output in MusicXML from each OMR program for every separate page for each score is combined to create a single MusicXML file for that score and that OMR program. As mentioned above, we chose to concentrate on the most important aspects of musical information, and so the post-processing steps described below ignored all grace notes and all elements of the MusicXML which did not simply describe pitch and rhythm. Most of the processing is done using music21 [8], using its data structures rather than directly processing the MusicXML.

The most common errors in the output are incorrect rhythms and missing notes. Occasionally there are errors of pitch, most often resulting from a failure to correctly recognise an accidental. Rests are also often incorrectly recognised, leading to erroneous rhythms. Sometimes larger errors occur, such as the failure to recognise an entire staff (occasionally a result of curvature in the scan), and the failure to correctly recognise a clef, can lead to a large-scale propagation of a single error.

The aim of our post-processing of MusicXML outputs was to arrive at a single combined MusicXML output which contained a minimum number of errors. However, this aim was challenging to fulfil because it was not possible to determine, based on the MusicXML alone which details are correct and which are incorrect. Our general approach, following Bugge *et al.* [4] is to align the outputs and to use majority voting as a basis for deciding which details are correct and which are incorrect.

The first post-processing steps apply to music with more than one voice on a single staff (as is common in keyboard music). Different OMR programs organise their output in different ways and some reorganisation is necessary to ensure that proper matching can take place. The steps in this part of the process are:

a) **Filling gaps with rests**. In many cases, rests in voices are not written explicitly in the score, and the OMR software recognises rests poorly. Furthermore, while music21 correctly records the timing offsets for notes in voices with implied rests, MusicXML output produced from music21 in such cases can contain errors where the offset is ignored. To avoid these problems, we fill all gaps or implied rests with explicit rests so that all voices in the MusicXML contain symbols to fill the duration from the preceding barline.

b) **Converting voices to chords**. The same music can be written using chords in some editions but separate voices in others. (See Figure 2 for an example.) To allow proper comparison between OMR outputs, we convert representations using separate voices into representations that use chords.

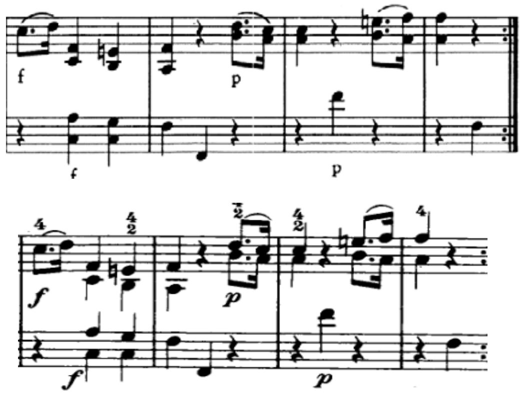

Figure 2. Extracts from the NMA and Peters editions of Mozart piano sonata K. 282, showing chords in the NMA where the Peters edition has separate voices.

c) **Triplets**. In many piano scores, triplets are common, but not always specified. Some OMR programs correctly recognise the notes, but not the rhythm. Our application detects whether the length of a bar (measure) matches with the time signature in order to determine whether triplets need to be inserted where notes beamed in threes are detected.

A grossly inaccurate output can lead to poor alignment and poor results when combining outputs. Therefore, it is better to exclude outputs which contain a lot of errors from subsequent alignment and voting, but again it is not possible to determine whether an output is grossly inaccurate on the basis of its contents alone. We once again employed the idea of majority voting: an output which is unlike the others is unlikely to be correct.

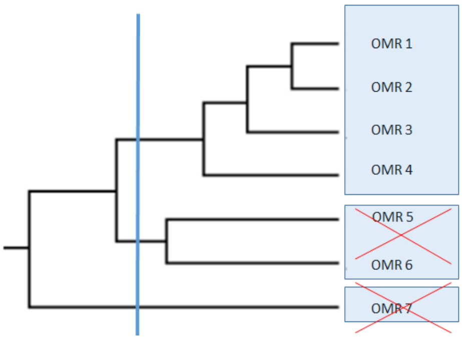

Figure 3. Example of phylogenetic tree with pruning of distant branches.

To determine how much outputs are like each other, we adopted the mechanism of 'phylogenetic trees' by UPGMA [24]. This mechanism is designed to cluster DNA sequences according to their similarity. Instead of a DNA string, each staff can be converted into a pitch-rhythm sequence and compared in pairs using the Needleman-Wunsch algorithm [15]. This process leads to the establishment of a similarity matrix and phylogenetic tree. Once the tree is configured, distant branches can be removed by a straightforward operation. So far, our best results have been obtained by using the three or four closest OMR outputs and discarding the rest. An evaluation of a more complex algorithm for pruning trees is left for future research.

3.3 Post-Processing: Alignment and Error-Correction

In order to determine the correct pitch or duration of a note, on the basis of majority voting, we need to know that the possible values in the different outputs refer to the same note. Therefore, it is necessary to align the outputs so that: the appropriate details in each output can be compared; a majority vote can be made; the presumably correct details can be selected; and a composite output can be constructed using these details.

There are two methods to align outputs, which we refer to as 'top-down' and 'bottom-up'. The first method aims to align the bars (measures) of the outputs so that similar bars are aligned. The second method aims to align the individual notes of the outputs irrespective of barlines. The first works better in cases where most barlines have been correctly recognised by the OMR software but the rhythms within the bars might be incorrectly recognised due to missing notes or erroneous durations. The second works better in cases where most note durations have been correctly recognised but the output is missing or contains extra barlines. In the following section, we explain the bottom-up process and how we combine the outputs from parts in order to obtain a full score. An explanation of our top-down approach used in earlier work can be found in [16].

3.3.1 Bottom-Up Alignment and Correction, Single Parts

Bottom-up alignment is applied to sequences of symbols from a single staff or a sequence of staves that correspond to a single part in the music. This might come from MusicXML output of OMR applied to an image of a single part in chamber music, such as a string quartet, full score, or keyboard music in which the staves for the right and left hands are separated. Each non-ignored symbol represented in the MusicXML (time signature, key signature, note, rest, barline, etc.) is converted to an array of values to give the essential information about the symbol. The first value gives the type of symbol plus, where appropriate, its pitch and/or duration class (i.e., according to the note or rest value, not the actual duration taking into account triplet or other tuplet indications). Alignment of OMR outputs, once again using the Needleman-Wunsch algorithm, is done using these first values only. In this way we are able to avoid alignment problems which might otherwise have occurred from one edition of a piece of music indicating triplets explicitly and another implicitly implying triplets, or from one OMR recognising the triplet symbol and another not recongising the symbol. All outputs are aligned using the neighbor-joining algorithm [20], starting with the most similar pair. A composite output is generated which consists of the sequence of symbols which are found to be present at each point in at least half of the outputs.

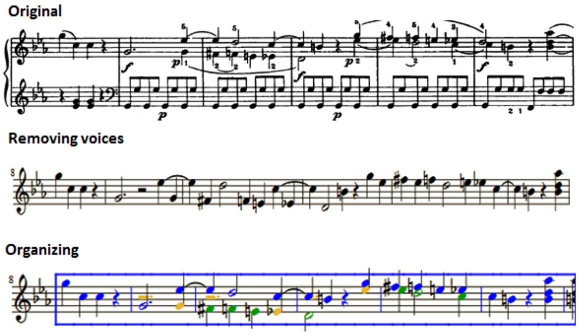

Figure 4. Example of removal of voices for alignment and subsequent reconstruction.

In the case of music where a staff contains more than one voice, such as in piano music, we adopt a procedure which creates a single sequence of symbols (see Figure 4). To achieve this, notes and rests are ordered first by their onset time. Next, rests are listed before notes and higher notes listed before lower notes. The result is a canonical ordering of symbols that make up a bar of multi-voice music. This means that identical bars will always align perfectly. Voice information (i.e., which voice a note belongs to) is recorded as one of the values of a note's array. However, this information is not taken into account when the correct values are determined by majority voting. Instead, voices are reconstructed when the aligned outputs are combined. Although, this means that the allocation of notes to voices in the combined output might not match any of the inputs (and indeed might not be deemed correct by a human expert), we found considerable variability in this aspect of OMR output and therefore could not rely upon it. At the same time, as in the pre-processing of MusicXML outputs from the OMR programs, additional rests are inserted into the combined output in order to ensure that every voice is complete from the beginning of each bar.

3.3.2 Top-Down Combined Output, Full Score

To generate a MusicXML representation of the full score, the results of alignment of the separate parts/staves need to be combined. Often the results for the constituent parts do not contain the same number of bars (measures), usually because of poor recognition of 'multi-rests' (i.e., several bars of rest that look like a single bar), or because of missing barlines. Sometimes OMR software inserts barlines where there are none. To achieve parts with the same number of bars and the best construction of the full score, the following steps are implemented:
a) **Finding the best full score OMR**. As a result of errors in recognising systems and staves, OMR is more likely to increase than reduce the number of bars. In the case of chamber music (e.g., string quartets), it is common to find OMR putting bars from one part in another part, adding extra rest bars and increasing the global number of bars. A simple algorithm is therefore used to select the OMR output that contains the smallest number of bars and the correct number of parts. We have found this to work correctly in most cases. As an example, the 724 bars in Mozart's string quartet K. 387 can be converted into 747, 843, 730 and 764 bars by different OMR programs. Figure 5 shows the result of OMR errors in interpreting the arrangement of staves, in this case causing two bars to be converted into four. One system of staves is mis-interpreted as two staves, and what is actually the second violin part is read as a continuation of the first violin part. In this case there is also the very common error of misreading the alto clef.

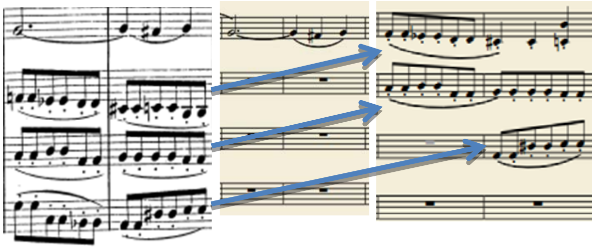

Figure 5. Displacement of parts in a string quartet.

b) **Aligning parts against the best full score**. Every bar of each part is converted into a sequence of hash values on the basis of the pitch and rhythm of the contents of the bar. The parts are aligned bar-by-bar (top-down approach) using the Needleman-Wunsch algorithm, and empty bars are introduced where needed. To determine the similarity of each pair of bars, the Needleman-Wunsch algorithm is used to align the contents of the two bars, and the aggregate cost of the best alignment taken as a measure of the similarity of the bars. For further detail, see [16]. This procedure results in correct vertical alignment of most of the bars and adds multi-rests that were not properly recognised in single parts.

3.4 Implementation

Our research was conducted using the Microsoft Windows operating system. This was because SharpEye only operates with this system. (The other three have versions for Windows and Macintosh systems.) Software was written in Python and made use of the Gamera and music21 libraries.

The items of OMR software used were designed for interactive use, so it was not a simple matter to integrate them into a single workflow. For this purpose we used the Sikuli scripting language.[1]

A cluster of six virtual machines running Windows Server 2012, controlled by a remote desktop connection, was setup to run the OMR software and our own pre- and post-processing software. To generate MusicXML from a set of scans of a piece of music, the required scans need to be stored in a particular directory shared between the different machines. The setup also provides different levels of Excel files for evaluating results.

With the exception of the commercial OMR programs, the software and documentation are available at https://code.soundsoftware.ac.uk/projects/multiomr and at http://github.com/MultiOMR.

4. EVALUATION

4.1 Materials

To test our system, we chose to use string quartets and piano sonatas by Mozart, because both scans and pre-existing symbolic music data for these pieces are available. The pieces tested were the string quartets dedicated to Joseph Haydn (K. 387, 421, 428, 458, 464 and 465) and the first six piano sonatas (K. 279, 280, 281, 282, 283 and 284). The sources have been taken from IMLSP (Peters edition, full scores and parts) and NMA (full scores). Ground truth files in Humdrum Kern format or MusicXML (when available) were downloaded from KernScores.[2] Two movements, (K. 428, mov. 4 and K. 464, mov. 1) are not yet complete on KernScores and so have not been evaluated. The string-quartet dataset included a total of 459 pages of music notation and the piano-sonata set 165 pages.

4.2 Results

For each piece, the output resulting from our system was compared with the data derived from KernScores and the

[1] www.sikuli.org
[2] kern.ccarh.org

recognition rate for notes was calculated (i.e., the percentage of notes in the original which were correctly represented in the output). Errors found in the output of our system compared to the ground truth were recorded in Excel and MusicXML files. These provided information about each incorrect note and its position in the score, the accuracy of the overall result, and the accuracy of each OMR program, plus colour-coding in the MusicXML file to indicate omissions, errors and insertions.

	OUT	CP	PS	SE	SS
Piano	95,11	74,26	86,13	91,86	85,40
String Quartet	96,12	47,84	81,47	86,65	82,40
Average	95,61	61,05	83,80	89,25	83,90

Table 1. Overall recognition rates. OUT = our system; CP = Capella-Scan; PS = PhotoScore; SE = SharpEye; SS = SmartScore.

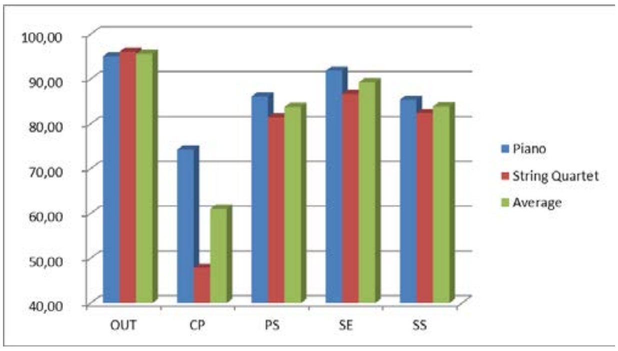

Figure 8. Overall recognition rates. For codes see the caption to Table 1.

Summary results are shown in Figure 8 and Table 1. As can be seen, overall, the output of our system was found to be better than all the OMR programs. The difference in recognition rates is higher for the string quartets due to very poor recognition in some cases for the Peters full score edition. One of the main strengths of our system rests with having consistent recognition rates of around 95% in most cases. This can be attributed to the automatic removal of scores poorly recognised at the phylogenetic-tree stage. This pruning removes a large amount of noise introduced by some OMR systems. A second strength is the lack of hard-coded rules. For instance, some OMR programs are better at recognising triplets and others at detecting pitch. Furthermore, the situation can change with new versions of OMR software and the introduction of new OMR programs. For our system to incorporate new versions and new OMR programs, all that is required is to add the necessary scripts in Sikuli to use those programs to generate MusicXML output from a given set of score images.

The entire process (six complete string quartets using parts and full scores), employing six virtualised machines in parallel, with an Intel Xeon processor of 3.30GHz, takes around 3 hours to complete.

5. FUTURE PROSPECTS

We have shown that by using pre-processing techniques and combining results from different OMR programs and different sources, we can significantly reduce the number of errors in optical music recognition. For those compiling corpuses of musical data in symbolic format, this would increase efficiency and save considerable effort. Furthermore, the reduction in error rate means that research which relies on large quantities of musical data which was previously impossible because it would be too costly to compile the data might now become possible. Large quantities of data can now be easily derived from scans available from IMSLP and elsewhere. Although errors remain in the output, the reduced error rate compared with raw OMR programs output will enable more valid results to be derived from statistical studies which previously lacked validity. Nevertheless, it should be noted that the number of errors which remain will still be too high for musicological research which assumes near 100% accuracy in the data.

Our research has made clear the limitations of current commercial OMR software. For example, in [16] we proposed the image pre-processing to extract separate bars in order to arrive at a better alignment of outputs that come from different OMR programs. However, we discovered that the OMR programs were generally incapable of producing any useful output from a single bar of music so we had to abandon this idea. It is our judgement, based on the project reported here, that further improvement of our system will require the limitations associated with current OMR software to be overcome. This may require a fundamentally different approach to currently available OMR programs.

Some possible directions have been proposed in the literature. For instance, Fahmy & Blostein [10] proposed a method based on rewriting graphs arising from raw symbol-recognition to produce graphs that conform more closely to musical constraints. Bainbridge & Bell [1] have also proposed making greater use of the semantics of musical notation. Raphael and co-workers [12, 17] have proposed a Bayesian approach based on likelihood, again taking account of musical constraints. Church & Cuthbert [7] proposed correcting errors by using information from similar musical sequences in the same piece of music. Finally, Rossant & Bloch [19] proposed the use of fuzzy logic. Since all of these approaches aim to take an image of a score and output the symbols which are represented in that score, they could be incorporated into the workflow described here, and its output combined with other outputs to further improve recognition.

6. ACKNOWLDGEMENTS

This work was supported with funding from the Arts and Humanities Research Council (AHRC; AH/L009870/1).

7. REFERENCES

[1] D. Bainbridge and T. Bell: "A music notation construction engine for optical music recognition," *Software—Practice and Experience*, Vol. 33, pp. 173–200, 2003.

[2] D. Bainbridge, C. G. Nevill-Manning, I. H. Witten, L. A. Smith, and R. J. McNab: "Towards a digital library of popular music," *Proc. of the 4th ACM conference on Digital libraries (DL '99)*, pp. 161–169, 1999.

[3] D. Bainbridge and K. Wijaya: "Bulk Processing of Optically Scanned Music", *Proc. of 7th Int. Conf. on Image Processing And Its Applications*, Vol. 1, pp. 474–478, 1999.

[4] E. P. Bugge, K. L. Juncher, B. S. Mathiasen, and J. G. Simonsen: "Using sequence alignment and voting to improve optical music recognition from multiple recognisers," *Proc. of the 12th Int. Soc. for Music Information Retrieval Conf.*, pp. 405–410, 2011.

[5] D. Byrd and M. Schindele: "Prospects for Improving OMR with Multiple Recognisers," *Proc. of the 7th Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 41–46, 2006. Revised and expanded version (2007) retrieved February 20, 2013, http://www.informatics.indiana.edu/donbyrd/MROMRPap.

[6] G. S. Choudhury, T. DiLauro, M. Droettboom, I. Fujinaga, B. Harrington, and K. MacMillan: "Optical Music Recognition System within a Large-Scale Digitization Project," *Proc. of the Int. Symp. on Music Information Retrieval*, 2000.

[7] M. Church and M. S. Cuthbert: "Improving Rhythmic Transcriptions via Probability Models Applied Post-OMR," *Proc. of the 15th Conf. of the Int. Soc. for Music Information Retrieval*, pp. 643–647, 2014.

[8] M. S. Cuthbert and C. Ariza: "music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data," *Proc. of the 11th Int. Soc. for Music Information Retrieval Conf.*, pp. 637–42, 2010.

[9] C. Dalitz, M. Droettboom, B. Pranzas, and I. Fujinaga: "A Comparative Study of Staff Removal Algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5), 753–766, 2008.

[10] H. Fahmy and D. Blostein: "A Graph-Rewriting Paradigm for Discrete Relaxation: Application to Sheet-Music Recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 12, No.6, pp. 763–799, 1998.

[11] M. Good: "MusicXML for notation and analysis,", in *The Virtual Score: Representation, Retrieval, Restoration (Computing in Musicology 12)*, W. B. Hewlett, and E. Selfridge-Field, Eds. MIT Press, 2001, pp. 113–124.

[12] R. Jin and C. Raphael: "Interpreting Rhythm in Optical Musical Recognition," *Proc. of the 12th Int. Soc. for Music Information Retrieval Conf.*, pp. 151–156, 2012.

[13] I. Knopke and D. Byrd: "Towards Musicdiff: A Foundation for Improved Optical Music Recognition Using Muliple Recognizers," *Proc. Of the 8th Int. Conf. on Music Information Retrieval*, pp. 123–126, 2007.

[14] H. Miyao and M. Okamoto: "Stave Extraction for Printed Music Scores Using DP Matching," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 8, No. 2, pp. 208–215, 2004.

[15] S. B. Needleman and C. D. Wunsch: "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of molecular biology*, Vol. 48, No. 3, pp. 443–453, 1970.

[16] V. Padilla, A. Marsden, A. McLean, and K. Ng: "Improving OMR for Digital Music Libraries with Multiple Recognisers and Multiple Sources," *Proc. of the 1st Int. Workshop on Digital Libraries for Musicology*, pp. 1–8, 2014.

[17] C. Raphael and J. Wang: "New Approaches to Optical Music Recognition," *Proc. of the 12th Int. Soc. for Music Information Retrieval Conf.*, pp. 305–310, 2011.

[18] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. Marcal, C. Guedes, and J. S. Cardoso: "Optical music recognition: state-of-the-art and open issues," *International Journal of Multimedia Information Retrieval*, Vol. 1, No. 3, pp. 173–190, 2012.

[19] F. Rossant and I. Bloch: "Robust and Adaptive OMR System Including FuzzyModeling, Fusion of Musical Rules, and Possible Error Detection," *EURASIP Journal on Advances in Signal Processing*, Vol. 2007, Article ID 81541, 25 pp.

[20] N. Saitou and M. Nei: "The neighbor-joining method: a new method for reconstructing phylogenetic trees," *Molecular biology and evolution*, 4(4), 406-425, 1987.

[21] C. S. Sapp: "Online Database of Scores in the Humdrum File Format," *Proc. Of the 6th Int. Conf. on Music Information Retrieval*, pp. 664–665, 2005.

[22] S. Suzuki and K. Abe: "Topological Structural Analysis of Digitized Binary Images by Border Following," *Computer Vision, Graphics and Image Processing*, 30(1), 32–46, 1985.

[23] V. Viro: "Peachnote: Music Score Search and Analysis Platform," *Proc. of the 12th Int. Soc. for Music Information Retrieval Conf.*, pp. 359–362, 2011.

[24] M. Zvelebil and J. Baum: *Understanding bioinformatics*. Garland Science, Abingdon, 2007.