IMAGE QUALITY ESTIMATION FOR MULTI-SCORE OMR

Dan Ringwalt, Roger B. Dannenberg
Carnegie Mellon University
School of Computer Science
ringwalt@cmu.edu, rbd@cs.cmu.edu

ABSTRACT

Optical music recognition (OMR) is the recognition of images of musical scores. Recent research has suggested aligning the results of OMR from multiple scores of the same work (multi-score OMR, MS-OMR) to improve accuracy. As a simpler alternative, we have developed features which predict the quality of a given score, allowing us to select the highest-quality score to use for OMR. Furthermore, quality may be used to weight each score in an alignment, which should improve existing systems' robustness. Using commercial OMR software on a test set of MIDI recordings and multiple corresponding scores, our predicted OMR accuracy is weakly but significantly correlated with the true accuracy. Improved features should be able to produce highly consistent results.

1. INTRODUCTION

Optical music recognition (OMR) is the problem of converting scanned music scores into a symbolic format such as MIDI. The advantages of OMR for computer music applications are clear, but it has yet to be widely used in many applications which use MIDI or MusicXML scores. Although OMR has been studied extensively since the 1960s, no OMR system has near-perfect accuracy. Commonly, the output of an OMR system must be checked by hand and at least a few corrections must be made, making the process extremely time-consuming [2]. This limits the amount of music which may be digitized, and in fact, much music is still digitized completely by hand in sources such as the Mutopia Project [1]. Recent research has focused on using contextual information beyond what is present on a single page to improve OMR results.

Recently, the Petrucci Music Library (or International Music Score Library Project, IMSLP) [17] has become a high-quality source of public domain music scores. The site allows users to scan and upload scores. Therefore, there may be several scores of the same work, which may be musically identical, or different editions, arrangements, or parts. There is a large discrepancy between the scanning equipment each user has, along with their relative care in scanning, so image quality varies widely. At the time of writing, IMSLP contains over 90,000 works, for which there are over 300,000 uploaded scores.

Although many OMR errors are due to notational complexity [2], we expect at least some mistakes to be due to random deformation in the score, independent of the content. Then if multiple scores are available corresponding to one piece, a consensus built from OMR applied to each score should be more accurate than any one score. The possibility of aligning multiple scores of the same work to build a single result (multi-score OMR, MS-OMR) is already being explored [21].

However, scores available from IMSLP and other sources vary widely in noise introduced in the scanning process. Previous work on multi-recognizer OMR (MR-OMR), where the results are aligned from several OMR systems on the same score, has noted that a consensus result using simple voting may be worse than the result of the best recognizer [4]. Similarly, if there are several poor scores for a work and one good score, a MS-OMR result may be worse than the result on the highest-quality score alone. An MS-OMR system that correctly estimates the quality of each score and acts accordingly should overcome this limitation.

Formally, we want to predict some accuracy measure of OMR, given features extracted from an image. We define the *quality* of an image to be the predicted accuracy given by our resulting model. Quality should depend on factors such as random noise, deformation of the page, and resolution, and is expected to be correlated with OMR accuracy. Our predicted value is mostly useful in comparisons between scores; even if the actual accuracy is on a 0 to 1 scale, a quality value learned using linear regression may be outside this range for some scores, and so it may not be interpretable as an accuracy value. However, even if we evaluate multiple recognizers using the same methodology, then we can learn a separate quality value for each recognizer, and predict the best-performing recognizer for a new score.

Clearly, the quality value gives useful information to a MS-OMR system. We may want to throw out some scores altogether if their quality is too low, as they may not contribute much of a benefit in addition to the higher-quality scores. As a simplification, we may only take the highest-quality score, and perform normal OMR. If our quality value is accurate, then this is the safest approach, because by introducing other scores, we risk lowering the accuracy.

This is clearly less computationally expensive than obtaining and aligning multiple OMR results, but should result in much higher accuracy than randomly choosing any available score. We consider this approach to MS-OMR in this paper.

2. RELATED WORK

2.1 Multi-Recognizer and Multi-Image OMR

Recent research has focused on improving OMR accuracy by aligning several OMR results and building a consensus score. Byrd and Schindele [5] designed a multi-recognizer OMR system which applies multiple OMR systems to the same score, and resolves conflicts between results using pre-defined rules. This is built on the assumption that each OMR system will have particular situations in which it outperforms the other systems. As OMR systems are under development and their strengths and weaknesses may change, a system is proposed which automatically learns the performance of each system in different possible situations.

More recently, Bugge et al. [4] proposed another multi-recognizer OMR system that resolves conflicts between each recognizer by a simple majority vote. Scores are exported as MusicXML from each recognizer, and converted to a custom subset of MusicXML, "MusicXiMpLe," which only stores the information necessary to decode note pitch and duration.

Padilla et al. have suggested extending multiple-recognizer OMR to align the results from multiple images of the same score [21]. A method is proposed to profile the response of each OMR tool to score quality, by adding additional noise to existing scores with available ground truth and measuring OMR accuracy.

2.2 Image Quality

Existing measures have been designed to estimate the level of degradation present in an image due to the scanning process. Kanungo et al. developed a local distortion model (referred to as *Kanungo noise*) for binary images which is an extension of simple salt-and-pepper noise, and uses 6 parameters [15]. The additional parameters capture the increased noise near the boundary between black and white pixels, and correlation in noise between nearby pixels.

Kanungo et al. previously estimated the Kanungo noise parameters of a binary image of a text document [16]. The estimation requires an ideal set of synthetic text documents with similar font face and size to the scanned image. Given an estimated set of parameters, each ideal image is degraded using the parameters. All 3x3 square patterns are found in each degraded ideal image and the input image, and a histogram for the count of each of $2^{3*3} = 512$ patterns is made for the degraded ideal images and the input. A Kolmogorov-Smirnov test statistic is measured between the cumulative distribution functions of both histograms. This statistic is minimized using the Nelder-Mead simplex method [19].

Additionally, prior work in OMR has focused on undoing global distortions present in the input image. The level of distortion detected by these methods is another feature which should be negatively correlated with OMR accuracy. For example, Fujinaga's staff detection algorithm [12] tries to correct bending of the staves due to page curl. This *deskewing* process translates each column of the image to make the staff lines more horizontal. We use the mean vertical translation performed by deskewing as one feature.

We may also robustly estimate the resolution of an image using the distance between staff lines. Unlike the actual size of the image, this does not depend on the size of the original page, and all symbols such as notes will be directly proportional to the staffline distance. We use Cardoso et al.'s robust estimated staffline distance [7] as another feature.

3. METHODS

3.1 Data Acquisition

All available scores of Ludwig van Beethoven's piano sonatas were obtained from IMSLP. In total, there were 32 sonatas, with 285 different scores.

MIDI versions of several movements from the Beethoven piano sonatas were obtained from the Mutopia Project [1], and served as ground truth to compare with the OMR results. The MIDI version was automatically generated from a manually transcribed LilyPond [20] source file.

As the MIDI files are separated by movement, the scores were also split into each movement. Therefore, each *work* is defined to be a single movement of a sonata.

3.2 Score Preprocessing

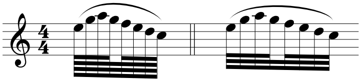
The scores were preprocessed by a custom system before extracting image quality features and performing OMR. Our methods for rotation correction and staff and staff system detection are described in [25].

Many scores had movements which started in the middle of the page. Therefore, the staff systems which formed the start of each movement were labeled by hand. Our system was used to automatically segment pages as necessary to split the score into movements.

We kept 67 original scores from IMSLP which contained an entire sonata and were not an arrangement or other version, and had ground truth for at least one movement available from the Mutopia Project. We successfully generated and processed 95 single-movement scores for 16 works (single movements), belonging to 8 different sonatas.

3.3 Image Quality Features

Kanungo parameter estimation was performed on each preprocessed page. A page from a LilyPond-engraved score obtained from the Mutopia Project was used as the ideal image. Each image was scaled to a normalized staffline



(a) Original score (b) Reconstructed score

Figure 1. This error only counts as one error under low-level evaluation, but several under high-level evaluation, as the length of each note is incorrect. Source: [24]

distance value of 8. Kanungo noise parameters were estimated using the SciPy [13] implementation of Nelder-Mead optimization [19], as described in Section 2.2. Nelder-Mead was run 10 times starting from a uniformly random parameter distribution, and was stopped after 50 function evaluations each time. The resulting Kanungo parameters $(\nu, \alpha_0, \alpha, \beta_0, \beta, k)$ were used as features to predict OMR performance.

We also performed Fujinaga's staff detection algorithm, which skews the image to correct page curl. This gives us the amount of page curl in the original image. We use the mean vertical translation performed by this deskewing as one feature, which represents the degree of distortion in the page.

Finally, we used Cardoso et al.'s robust staffline distance estimation method [7]. We used the staffline distance, and the ratio of staffline thickness to distance, as two more features. The staffline distance represents the resolution of the image, while the thickness-to-distance ratio represents the relative thickness of lines on the page.

3.4 OMR

The preprocessed movements were processed by the SharpEye 2 OMR system, version 2.68. The result was exported to MIDI.

4. EVALUATION

4.1 OMR Evaluation Methods

OMR researchers have yet to adopt any evaluation metric as a common standard [6], and specialized evaluation methods will likely be needed for most systems. We chose as basic of an evaluation method as possible: simply comparing the start time of each note to the ground truth. This still requires rests, accidentals, and other basic symbols to be detected correctly in the usual case; it cannot detect a too-short note followed by a too-long rest, but this particular error should be extremely rare. Although it does not test other information like dynamic markings, we consider these to be of secondary importance compared to the actual notes. As we only consider the start position of each note, and not the duration of notes and rests, our evaluation is a further simplification of previous evaluations, which consider both the start and end of notes [4, 14].

SharpEye 2 outputs a proprietary .mro format which contains information such as the position of some individual symbols. Therefore, it is possible to conduct a *low-level* evaluation if the score is labeled with the position of each symbol. Although both values should be highly correlated, high-level accuracy may decrease drastically with only a small decrease in low-level accuracy, as illustrated in Figure 1.

Our evaluation method is considered high-level. This allows us to use MIDI recordings from the Mutopia Project, which only contain the actual notes, as our labeled data. One potential issue with MIDI is that to simulate a realistic performance, staccato notes may have a shortened length followed by a rest for their remaining time. Our evaluation, which only tests the start of each note, accounts for this.

4.2 Accuracy Value

Given two aligned scores, we need to derive a single value for the accuracy. Here, each note is represented as the time in the score, and a pitch, and a note is correctly detected if there is a note with the exact same values in the original score. The OMR output may contain both false positives, where a note is accidentally detected, and false negatives, where a note is missing. We may calculate the precision p, which is the proportion of true positives to all detected notes, and the recall r, which is the proportion of true positives to all notes in the original score. The standard method of combining these values, which we use as our accuracy value, is the F_1 score:

$$F_1 = \frac{2pr}{p+r}$$

4.3 MIDI-MIDI Alignment

All MIDI files were imported into Python using music21[8]. Next, we aligned each OMR output to the ground truth, to correct for missing or extra measures due to OMR errors. We noticed that LilyPond's MIDI output (used by Mutopia) pads a pickup measure to the length of a full measure, while SharpEye 2's does not. Therefore, we align each beat rather than each measure, so that the pickup will also be correctly aligned.

The standard alignment algorithm, used in both bioinformatics and computer music applications, is Needleman-Wunsch [18, 3]. It minimizes the sum of the distance between each aligned element of two sequences, plus a penalty for each inserted gap. In our case, our distance matrix has one row for each beat in the real score, and one column for each beat in the OMR score. The distance entry for each pair is $1 - F_1$ for the pair of beats, multiplied by the maximum of the number of notes in both beats. (This is implicitly 0 when both beats only contain rests, and the F_1 score would normally be undefined.) We use a gap penalty of 10.

After Needleman-Wunsch, we simply calculate the F_1 score for the entire aligned scores, with new positions for the notes accounting for inserted gaps. This is our OMR accuracy value.

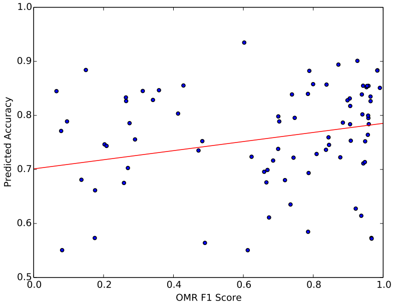


Figure 2. The OMR F_1 score for each movement compared with the predicted accuracy, with the best-fit line.

4.4 Quality Estimation

We used a linear model to predict the OMR F_1 score given our features. We chose the Scikit-learn [22] implementation of Support Vector Regression with a linear kernel, as it seemed to perform better than ordinary least squares linear regression. The model was validated by leave-one-out testing on each work: for each work, a model was trained excluding its corresponding scores, and the predicted best score for the work was compared to the score with the highest accuracy. Finally, we fit the model to the entire dataset to determine the coefficients.

5. RESULTS

The OMR F_1 score and predicted accuracy were weakly but significantly correlated ($R = 0.30, p = 0.0029$). The data is shown in Figure 2.

For each work, we compared the score with the highest OMR accuracy and the score with the highest predicted quality using leave-one-out testing (Table 1). Six of the 16 works had a correctly predicted best score, whereas using uniformly random guessing, the expected number of correct scores is only 2.82. The full OMR accuracy results are presented in Table 2.

We also noted that the best few scores may all have nearly the same high accuracy. In these cases, it is not necessary that our top predicted score has the highest accuracy, but the accuracy should be close to the highest. For each work, we considered the mean accuracy of all scores, which is the expected accuracy of a score selected by random choice, the highest accuracy, and the accuracy of the predicted best score. The mean of the expected accuracy for random guessing is 0.61, and the mean of the best accuracy (the best possible result) is 0.82, while the mean accuracy of the best predicted scores is 0.74. The chosen score's accuracy was higher than expected in 14 of 16 cases. This confirms that our method reliably outperforms random guessing, but there is still room to improve in choosing one of the best scores.

Work	Best	Pred. Best	# Scores
1.1	IMSLP66390	IMSLP05524	8
1.4	IMSLP66390	IMSLP05524	7
5.1	**IMSLP66394**	**IMSLP66394**	5
5.3	**IMSLP66394**	**IMSLP66394**	5
6.3	**IMSLP66395**	**IMSLP66395**	5
19.1	IMSLP00019	IMSLP04073	6
19.2	IMSLP05545	IMSLP69581	6
20.1	IMSLP45469	IMSLP66410	7
20.2	**IMSLP05546**	**IMSLP05546**	7
23.2	IMSLP51795	IMSLP04078	3
23.3	**IMSLP66412**	**IMSLP66412**	6
25.1	**IMSLP66414**	**IMSLP66414**	6
25.2	IMSLP66414	IMSLP69588	6
25.3	IMSLP66414	IMSLP69588	6
27.1	IMSLP66416	IMSLP05553	6
27.2	IMSLP69590	IMSLP05553	6

Table 1. Accuracy predictions on the Beethoven piano sonata test set. For each work (identified by *sonata number.movement*), we compare the score with the highest OMR accuracy (*Best*) and the highest predicted quality (*Pred. Best*).

The coefficients of our linear model (Table 3 in the appendix) are directly interpretable as the effect each parameter has on OMR accuracy. Many results were unexpected. For example, ν represents the probability of salt-and-pepper noise in the Kanungo model, which should negatively affect OMR accuracy, but its coefficient is positive. However, as it is on a small scale (typically $0 - 0.05$), it has a smaller impact on accuracy. This result may be due to a few outliers which had poor results for Kanungo estimation.

The coefficient for mean_skew, which is the deformation undone by Fujinaga's deskewing, is also unexpectedly positive. This may indicate a flaw in our implementation, or again, outliers. We did find that staff_dist is positively correlated with accuracy, as we expect that higher-resolution scores will have better results. The coefficient is small, but more significant as staff_dist is on a larger scale (usually at least 20).

6. CONCLUSIONS

We introduced an estimated OMR accuracy measure, and showed that its correlation to the true accuracy is statistically significant. However, the correlation is too low to correctly predict the best-quality score a majority of the time. On the other hand, this validates the use of features extracted from the image to select higher-quality scores. By refining our features and adding additional ones, we should be able to build a practical quality estimation system which can support multi-score OMR.

Since most of our current image features are parameters for Kanungo noise, the success of the image quality estimation is dependent on these parameters being accurate. The Kanungo estimation process is currently very time-intensive, requiring around 2 minutes per page. Some instances of Nelder-Mead will become stuck in a local optimum, so repeating Nelder-Mead even more times should improve results. However, the time involved made this impractical in this case.

7. FUTURE DIRECTIONS

7.1 Image Quality Features

We only found a weak correlation between OMR accuracy and predicted quality, and noted that the Kanungo parameter estimates were noisy. Furthermore, the estimation process is too slow to be practical for a large music library such as IMSLP. Therefore, a better performing, faster Kanungo estimation process is needed to make image quality estimation practical.

We may be able to improve Kanungo estimation by using assumptions specific to music scores, which would allow us to test a much smaller area of the image. For example, if we find all empty stretches of staff on the page, we can concatenate some of these as the input to Kanungo estimation. We may generate an ideal empty staff using the estimated staffline distance and thickness. This uses a much smaller image, and may even be more robust as differences in typography between the ideal and input image will not affect it.

Finally, our features only take into account errors introduced in the scanning process. However, differences in the original score, such as different fonts, should also affect the accuracy of a particular OMR recognizer. *Adaptive* OMR systems [11, 23] improve their performance on scores with a certain font and other particularities by learning from their corrected output. If an adaptive system is trained using a homogenous set of scores with a particular font, then we may be able to extract information about the font from its classification model. Features which have been used for handwritten music writer identification [10] may be useful.

7.2 OMR Evaluation

We mentioned that a small difference in low-level accuracy may make a dramatic difference in high-level accuracy. Therefore, low-level accuracy may be a more stable value to use when performing regression. However, obtaining a real-world test set of a similar size with low-level ground truth would be much more time-consuming.

Using scores from the Mutopia Project, it would be possible to modify LilyPond to output the position of each symbol, giving us a low-level ground truth. Next, we could apply deformations such as Kanungo noise to the output before performing OMR. This is similar to Padilla et al.'s proposal to add additional noise to real images from IMSLP to profile each OMR recognizer. However, if we start from ideal computer-engraved images, then the parameters we use to add noise to the image are exactly the same as our image quality features. Therefore, we may design our test set to cover the entire parameter space, and we can directly learn our image quality function using regression from the input parameters to the OMR accuracy for each recognizer.

On the other hand, we may be able to improve our results while keeping high-level accuracy. We may obtain a broader range of scores from IMSLP paired with MIDI recordings from the Mutopia Project, which would provide us with more training data. Using more data, we could train a more sophisticated model than linear regression, which would hopefully better predict accuracy. We noted that a single error has a proportional effect in low-level accuracy but a much bigger effect on high-level accuracy, so high-level accuracy likely has a nonlinear relationship with quality. Therefore, methods such as kernel SVR or random forests may be able to capture this nonlinear relation.

We noticed that some MIDI scores were unable to be opened by music21, and they were excluded from the analysis. This is believed to be because some note durations cannot be unambiguously converted from a floating-point time value back to the music-theoretic note values which music21 uses. This should be possible to fix by using the MIDI files in their original form, which would allow us to include more data in our analysis.

7.3 Alignment-Based MS-OMR

Although we presented our method as a simpler alternative to existing MS-OMR systems, our image quality estimate may be used in a larger system. An MS-OMR system which aligns multiple results, as in [21], may be augmented by weighting each score by its quality in the vote. Furthermore, alignment-based MS-OMR systems require a multiple sequence alignment, and finding the globally optimal such alignment is NP-complete [26]. Approximate multiple alignment algorithms often use a series of pairwise alignments [9]. Recent research in aligning multiple musical recordings or scores used a progressive alignment, where pairwise alignments were performed sequentially on the inputs [27, 4]. Ordering OMR results from highest to lowest quality may work better than other orders.

We have demonstrated the usefulness of image quality estimation in predicting OMR accuracy. A more robust quality estimate should be useful for any MS-OMR system. This should have a significant impact on OMR accuracy for large music libraries such as IMSLP.

Work	Score	Accu.	Qual.	Score	Accu.	Qual.	Score	Accu.	Qual.	Score	Accu.	Qual.	Score	Accu.	Qual.
1.1	00001	0.95	0.62	03796	0.70	0.79	05524	0.95	0.88	51707	0.57	Error	66279	0.70	0.75
1.1	66390	0.96	0.79	77993	0.61	0.50	90564	0.08	0.18	243106	0.84	0.83			
1.4	00001	0.94	0.56	03796	0.79	0.66	05524	0.31	0.82	51707	0.37	Error	66279	0.78	0.85
1.4	66390	0.96	0.86	77993	0.62	0.70	243106	0.67	0.65						
5.1	00005	0.92	0.82	02412	0.08	Error	03858	0.81	0.69	51714	0.85	Error	66394	0.96	0.83
5.1	68715	0.69	0.69	243114	0.91	0.75									
5.3	00005	0.17	0.67	03858	0.49	0.46	51714	0.18	Error	66394	0.60	0.87	68715	0.47	0.71
5.3	243114	0.17	0.53												
6.3	00006	0.41	0.79	03859	0.34	0.80	51715	0.40	Error	66395	0.43	0.85	68719	0.36	0.68
6.3	243121	0.10	0.80												
19.1	00019	0.75	0.72	04073	0.15	0.89	05545	0.26	0.76	45370	0.26	0.62	51743	0.20	0.73
19.1	66408	0.28	Error	69581	0.72	0.62	345618	0.27	Error						
19.2	00019	0.91	0.75	04073	0.84	0.74	05545	0.94	0.84	45370	0.94	0.78	51743	0.67	0.57
19.2	66408	0.98	Error	69581	0.93	0.93	345618	0.94	Error						
20.1	00020	0.08	0.66	04075	0.85	0.75	05546	0.96	0.83	45469	0.97	0.52	51745	0.67	0.54
20.1	66410	0.07	0.82	69582	0.90	0.85									
20.2	00020	0.95	0.59	04075	0.14	0.64	05546	0.98	0.88	45469	0.95	0.74	51745	0.79	0.50
20.2	66410	0.08	0.50	69582	0.94	0.70									
23.2	03184	0.11	0.51	04078	0.46	0.70	51795	0.55	0.59						
23.3	00023	0.57	0.67	03184	0.09	0.55	04078	0.38	0.72	05549	0.58	0.78	51795	0.41	0.53
23.3	66412	0.60	0.86												
25.1	00025	0.97	0.52	03185	0.43	Error	04081	0.80	0.83	05551	0.96	0.76	51797	0.88	0.68
25.1	66414	0.98	0.88	69588	0.26	0.84									
25.2	00025	0.84	0.73	04081	0.73	0.57	05551	0.95	0.85	51797	0.74	0.64	66414	0.99	0.68
25.2	69588	0.87	0.89												
25.3	00025	0.92	0.56	04081	0.66	0.63	05551	0.96	0.79	51797	0.74	0.70	66414	0.96	0.74
25.3	69588	0.94	0.84												
27.1	00027	0.88	0.75	04090	0.79	0.86	05553	0.90	0.83	51799	0.48	0.76	66416	0.91	0.81
27.1	69590	0.70	0.78												
27.2	00027	0.27	0.69	04090	0.21	0.70	05553	0.27	0.75	51799	0.18	0.60	66416	0.29	0.74
27.2	69590	0.51	0.55												

Table 2. OMR accuracy (F_1) values for each score (by IMSLP ID), and predicted quality values.

Variable	Coefficient	Variable	Coefficient
ν	4.2	mean_skew	19.34
α_0	1.7	staff_dist	0.021
α	0.10	staff_thick_ratio	0.22
β_0	-0.70		
β	-0.077		
k	-0.0026		

Table 3. Coefficients of the linear model for image quality.

8. REFERENCES

[1] The Mutopia Project, 2015. http://mutopiaproject.org/ (accessed July, 2015).

[2] D. Bainbridge and T. Bell. The challenge of optical music recognition. *Computers and the Humanities*, 35(2):95-121, 2001.

[3] J. J. Bloch and R. B. Dannenberg. Real-time accompaniment of polyphonic keyboard performance. In *Proceedings of the 1985 International Computer Music Conference*, pages 279-290, 1985.

[4] E. P. Bugge, K. L. Juncher, B. S. Mathiasen Jakob, and J. G. Simonsen. Using sequence alignment and voting to improve optical music recognition from multiple recognizers. In *Proceedings of the 12th International Conference on Music Information Retrieval*, pages 405-410, 2001.

[5] D. Byrd and M. Schindele. Prospects for improving OMR with multiple recognizers. In *Proceedings of the 7th International Conference on Music Information Retrieval*, pages 41-46, 2006.

[6] D. Byrd and J. G. Simonsen. Towards a standard testbed for optical music recognition: definitions, metrics, and page images, 2015. http://www.informatics.indiana.edu/donbyrd/OMRTestbed/OMRStandardTestbed1Mar2013.pdf (accessed July, 2015).

[7] J. S. Cardoso and A. Rebelo. Robust staffline thickness and distance estimation in binary and graylevel music scores. In *20th International Conference on Pattern Recognition*, pages 1856-1859, 2010.

[8] M. S. Cuthbert and C. Ariza. music21: A toolkit for computer-aided musicology and symbolic music data. In *Proceedings of the 11th International Conference on Music Information Retrieval*, pages 637-42, 2010.

[9] R. C. Edgar. MUSCLE: Multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, 32(5):1792-7, 2004.

[10] A. Fornés, A. Dutta, A. Gordo, and J. Lladós. The ICDAR 2011 music scores competition: staff removal and writer identification. In *2001 International Conference on Document Analysis and Recognition*, pages 1511-15, 2011.

[11] I. Fujinaga. *Adaptive Optical Music Recognition*. PhD thesis, McGill University, 1996.

[12] I. Fujinaga. Staff detection and removal. In *Visual Perception of Music Notation: On-Line and Off-Line Recognition*, pages 1-39, 2004.

[13] E. Jones, T. Oliphant, P. Peterson, *et al.* SciPy: Open source scientific tools for Python, 2001. http://www.scipy.org/ (accessed July, 2015).

[14] G. Jones, B. Ong, I. Bruno, and K. Ng. Optical music imaging: music document digitisation, recognition, evaluation, and restoration. In *Interactive Multimedia Music Technologies*, pages 50-79, 2008.

[15] T. Kanungo, R. M. Haralick, and I. Phillips. Global and local document degradation models. In *Proceedings of the Second International Conference on Document Analysis and Recognition*, pages 730-734, 1993.

[16] T. Kanungo and Q. Zheng. Estimation of morphological degradation model parameters. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing Proceedings*, volume 3, pages 1961-1964, 2001.

[17] Project Petrucci LLC. IMSLP/Petrucci Music Library, 2015. http://imslp.org/ (accessed July, 2015).

[18] 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*, 48(3):443-453, 1970.

[19] J. A. Nelder and R. Mead. A simplex method for function minimization. *The computer journal*, 7(4):308-313, 1965.

[20] H.-W. Nienhuys and J. Nieuwenhuizen. LilyPond, a system for automated music engraving. In *Proceedings of the XIV Colloquium on Musical Informatics* (XIV CIM 2003), pages 1-6, 2003.

[21] V. Padilla, A. Marsden, A. McLean, and K. Ng. Improving OMR for digital music libraries with multiple recognizers and multiple sources. In *Proceedings of the 1st International Workshop on Digital Libraries for Musicology - DLfM '14*, pages 1-8, 2014.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825-2830, 2011.

[23] L. Pugin, J. A. Burgoyne, and C. Ha. MAP adaptation to improve optical music recognition of early music documents using hidden Markov models. In *Proceedings of the 8th International Conference on Music Information Retrieval*, pages 513-16, 2007.

[24] T. Reed. *Optical music recognition*. Master's thesis, University of Calgary, 1995.

[25] D. Ringwalt, R. B. Dannenberg, and A. Russell. Optical music recognition for live score display. In *Proceedings of the 2015 Conference on New Interfaces for Musical Expression*, 2015.

[26] L. Wang and T. Jiang. On the complexity of multiple sequence alignment. *Journal of Computational Biology*, 1(4):337-348, 1994.

[27] S. Wang and S. Dixon. Robust joint alignment of multiple versions of a piece of music. In *Proceedings of the 15th International Conference on Music Information Retrieval*, pages 83-88, 2014.