Human-Guided Recognition of Music Score Images

Liang Chen
Indiana University Bloomington
Bloomington, Indiana
chen348@indiana.edu

Rong Jin
Indiana University Bloomington
Bloomington, Indiana
rongjin.jr@gmail.com

Christopher Raphael
Indiana University Bloomington
Bloomington, Indiana
craphael@indiana.edu

ABSTRACT

We present our ongoing work in optical music recognition in which we seek to transform printed music notation images into symbolic representations, suitable for playback, analysis, and rendering. While music notation contains a small core of symbols and primitives composed in a rule-bound way, there are a great many common exceptions to these rules, as well as a heavy tail of rarer symbols. Since our goal is to create symbolic representations with accuracy near that of published music scores, we doubt the feasibility of fully-automatic recognition, opting instead for a human-guided approach. We define a simple communication channel between the user and recognition engine, in which the user imposes pixel-level or model-level constraints.

CCS CONCEPTS

•Theory of computation →Interactive computation;

KEYWORDS

Optical Music Recognition, Human-in-the-loop Computation, Constrained Optimization

ACM Reference format:
Liang Chen, Rong Jin, and Christopher Raphael. 2017. Human-Guided Recognition of Music Score Images. In *Proceedings of 4th International Workshop on Digital Libraries for Musicology, Shanghai,China, October 28,2017 (DLfM '17),* 4 pages.
DOI: 10.1145/3144749.3144752

1 INTRODUCTION

Optical Music Recognition (OMR) seeks to convert music score images into symbolic representations. Such symbolic music formats provide a hierarchical structured description of the music, including notions of part, measure, voice, and note, as well as detailed descriptions of the individual symbols. Success with OMR would pave the way for large symbolic libraries containing all the world's public domain music, that could be instantly accessed, searched, transformed, and reformatted. Such libraries would provide a greatly improved experience for musicians through digital music stands; it would serve as the backbone for developing academic fields, such as computational musicology; finally, it would enable emerging applications fusing music and computation such as data-driven composition systems, musical accompaniment systems, and automatic music transcription.

OMR research dates back to the 1960s with mostly-disconnected approaches to many aspects of the problem [1, 9, 12–15, 19] including several overviews [3, 16], and well-established commercial systems [20, 21]. In spite of these efforts there is still much to accomplish before the sought-after large-scale symbolic libraries will be in reach. The reason is simply that OMR is *hard*, constituting, in our view, one of the grand challenges of document recognition.

Part of OMR's difficulty lies in the high degree of necessary recognition accuracy. The future's digital music stands will require accuracy at least as good as the familiar published hard-copy scores they will displace, otherwise this new technology will not be embraced. Music performance and scholarship require faithful and precise data, thus OMR researchers must frame the ultimate goals with the musical community's standards in mind.

In addition to the high bar regarding quality, OMR poses significant technical challenges. One such difficulty arises from the *heavy tail* of standard music symbols: while a small core of symbols account for most of printed ink, these are complemented by a long list of familiar symbols absent from many or most pages. While each of these symbols may be recognized by fairly standard means, they are rare enough that their inclusion often results in more false positives than correct detections. This exemplifies of one of OMR's most vexing challenges: we cannot simply ignore unusual symbols, though their inclusion often leads to worse recognition performance.

Analogous to this *rare-symbol problem* are cases in which core symbols are used in unusual ways — nearly all of music notation's "rules" are broken occasionally. For example, note heads usually lie on one side of the stem, though densely-voiced chords usually result in "wrong side" note heads; beamed groups usually have closed note heads though tremolos and certain "double-duty" usages leads to open note heads in a beamed group; beamed groups are usually associated with a single staff, but can span both staves of a grand staff when necessary. As with the heavy tail of symbols, these special cases can all be modeled and recognized, though the result is often worse performance than a more restrictive approach would produce. Again we see the same essential paradox for OMR system construction. While we doubt the value of an OMR approach that does not account for these unusual-but-not-rare special cases, their inclusion often results in degraded performance overall.

Given the demand for high accuracy and the technical challenges mentioned, we are skeptical that any fully automatic OMR approach will ever deal effectively with the wide range of situations encountered in real-life recognition scenarios. Other OMR reserchers have reached similar conclusions [17] suggesting interactive OMR as a more realistic solution to the problem. We formulate

DLfM '17, Shanghai,China
© 2017 ACM. 978-1-4503-5347-2/17/10…$15.00
DOI: 10.1145/3144749.3144752

the challenge as one of *human-in-the-loop computing*, developing a *mixed-initiative system* [5, 10, 11] that fuses both human and machine abilities. To some extent a human-guided approach seems unavoidable; without the human "seal of approval," the value of the resulting music data will always be held in question, thus almost requiring an approach where the human plays *some* role. Since we must involve the human somewhat, it seems appropriate to make the most of this contribution.

In what follows we present our view of human-in-the-loop OMR. Our essential idea, presented in Section 3, is to allow the user two axes of control over the recognition engine. In one axis the user chooses the *model* that can be used for a given recognition task, specifying both the exceptions to the rules discussed above, as well as the relevant variety of symbols to be used. In the other, the user labels misrecognized pixels with the correct primitive or symbol type, allowing the system to re-recognize subject to these user-imposed constraints. This approach results in a simple interface in which the user can provide a wealth of useful knowledge without needing to understand the inner-workings and representations of the system. Thus we effectively address the *communication* issue between human and recognition system. Our work has commonalities with various other human-in-the-loop efforts such as [4, 22], though most notably with other approaches that employ constrained recognition as driven by a user [2, 6, 18].

Section 4 compares the output of our system with that of the commercial system, *SmartScore*, which we believe to be among the best commercial systems. The overall results show the systems to be comparable, requiring similar effort for a given level of accuracy. However, given the greater raw recognition accuracy of *SmartScore* over our current system (at present), the results demonstrate the value of our proposed human-in-the-loop contribution.

2 AUTOMATIC RECOGNITION

Due to the page limit, we leave out the details of our recognition model here, which has been elaborated in our previous work [7, 8].

Automatic recognition begins by first identifying the staves, and then grouping these staves together into systems, while estimating the common bar lines for each system. This preliminary phase provides the structure for nearly all subsequent recognition in which we treat the system measure as the basic unit of analysis.

For each system measure we perform independent searches for the various musical symbols contained in the measure. In doing so our goal is not to perform perfect recognition, but rather to prime the system for human-guided recognition. We hope to identify approximations of most of the symbols in the measure as well as to correctly segment the measure into distinct non-overlapping symbols. While the subsequent human-guided recognition of Section 3 can begin from scratch, the process is much more efficient when primed with partially correct results.

3 HUMAN-GUIDED RECOGNITION

The automatic recognition of Section 2 is done *off-line*, thus placing no demands on the user's time. For a given level of accuracy, we believe *user time* is the appropriate metric for OMR system performance, as this metric determines throughput of a user-guided system. Thus our goal is to create a system that allows the user to work as quickly as possible, while facilitating the user's ability to detect and correct errors.

After the automatic off-line recognition, the user is presented with the results for correction, displayed with transparent color drawn directly over the original image. This allows for straightforward comparison with the recognized results, in which both unwanted and missing recognized "ink" are easy to perceive. This approach contrasts sharply with the design choice of several commercial systems that display original and recognized images side-by-side. Finding discrepancies between these paired images poses a difficult and tiring cognitive challenge.

During the human-guided process, the user explores the page, measure-by-measure, looking for errors. Each pitch-bearing symbol can be played, either in isolation or in the entire measure, thus allowing one to *hear*, as well as *see*, the results. The user toggles between the symbols in a measure by first selecting one of the possible categories: *beam, chord, key-clef-signature, isolated symbol, slur, etc.*, then toggling through the symbols belonging to the category. The system highlights the currently-selected symbol, as well as displaying the associated candidate. During this process a completely incorrect symbol can simply be deleted. More often, the user will choose a symbol that is *partially* correct, proceeding to address the errors, as follows.

In correcting recognition errors, our system allows two axes of control. The first of these is composed of a collection of "switches" specifying the recognition grammar to be used. Whenever the switch settings are changed, the graph representing the relevant grammar is recomputed. The user then re-recognizes the current symbol according to the new switch settings. Many recognition problems can be corrected only using this axis of control, while switch settings are often appropriate for entire regions of the page.

The other axis of control allows the user to label individual image pixels with the symbol (e.g. sharp, treble clef, quarter rest, slur, augmentation dot, etc.) or primitive (e.g. open note head, ledger line, stem, double beam, stem, etc.) that covers the pixel. The user can re-recognize as many times as is needed, simultaneously adding pixel labels and changing the switch settings until the desired result is achieved.

We include a real-time movie[1] showing the real-time use of the system on a page from our test set. One can get an informal sense of the process and its efficiency from this video, though Section 4 takes a more empirical approach to evaluating our system.

4 EVALUATION

We compare our system with the commercial OMR system, SmartScore, in terms of both efficiency and accuracy. While we have not formally compared the various commercial systems, our subjective impression is that SmartScore is, at least, one of the best. SmartScore's raw symbol recognition ability is impressive, handling a wide range of symbol variations with high accuracy. While we cannot directly measure intermediate results, SmartScore's raw recognition accuracy appears to be significantly better than ours (at present). However, SmartScore takes less care with the correction of symbols, which holds the system back in an important way. That is,

[1] https://drive.google.com/file/d/0Bym22ztgqpjOeTltSXpQVWJlZjQ/view

this system appears to be conceived primarily in terms of automatic recognition, with an afterthought that allows the user to correct individual errors. In contrast, our system was conceived as a human-in-the-loop system.

For both systems we concentrate on what happens *after* automatic recognition, focusing exclusively on the human effort necessary to correct the results. We record the user time, mouse clicks, and key strokes, necessary to produce the resulting symbolic data, summarizing both the user effort and resulting accuracy for both systems. We focus on user time since the need for high accuracy means that user time will, most likely, be the bottleneck in large-scale deployment of OMR systems. In assessing accuracy, we view the result of each system as a collection of notes, with pitch, onset time, and duration for each note — these are easily computed from the MIDI data output by SmartScore, and are easy to compute from our system as well.

In SmartScore, proofreading and correction are the only aspects of the user effort we measure, since there is no user-guided recognition — the user simply adds, deletes, or changes the recognized symbols to fix pitch, timing or voice problems. In our system, the user corrects errors by imposing pixel-level and model-level, while the system re-recognizes subject to these constraints. While beyond the scope of the current paper, we applied an analogous approach to the correction of errors in rhythmic interpretation, recognizing rhythm subject to individual note constraints and model choices made by the user.

Our image data consist of 15 pages from 3 different piano pieces: 6 of the pages are from Frédéric Chopin's Fantasie Impromptu (C.F. Peters 1879 edition), 5 pages are the first movement of W.A. Mozart's Piano Sonata No.13 in B flat major (Breitkopf and Härtel 1878 edition), and 4 pages are the Rigaudon of Maurice Ravel's Le Tombeau de Couperin (Durand and Cie. 1918 edition)[2]. Overall, these appear to be medium-difficulty piano scores. In many ways piano music poses the greatest challenge for OMR, though we believe it is also the single most important area due to the wide repertoire for piano, as well as the existence of piano transcriptions for so much non-piano music.

Our evaluation of accuracy only considers the resulting pitch and rhythm. Clearly there is much more to OMR besides these two aspects; however, evaluating the accuracy of other symbols, such as dynamics, text, articulations, etc. require access to the internal representations of the OMR system. As we don't have that for SmartScore, we measure instead the results on the note-level results, obtained through the MIDI output. One problem with note-level evaluation is that single symbol recognition errors may produce many note errors: consider the effect of missing a clef change or note duration in a long measure. We have endeavored to make sure that the human correction accounts for these kinds of mistakes.

We view each note as a triple, (o, d, p), where o denotes the note onset time in quarters, d is the duration of the note, and p represents MIDI numbering of the pitch. A pair of notes produced by each of the two systems "match" if and only if all the attributes are equal. We excluded trills and grace notes in our experiments as the rhythmic interpretation of these are largely arbitrary. With both systems, the human-directed portion concentrates on the notational aspects relating to pitch and rhythm output, ignoring, for instance, text, articulations, and dynamics. While this excludes some important facets of the notation from our evaluation, the overwhelming majority of symbols relate to pitch and rhythm, so we believe the evaluation is still meaningful.

		Events	FP	FN	Time (min)
Chopin	User **S**	504	5.5	6.8	12.8
	User **A**	507	8.5	7.5	7.2
	User **B**	507	7.2	6.3	8.4
Mozart	User **S**	457	5.2	6.0	8.7
	User **A**	459	6.1	4.8	6.1
	User **B**	458	11.8	10.8	4.0
Ravel	User **S**	457	3.5	5.0	7.3
	User **A**	458	8.0	8.2	8.2
	User **B**	459	7.2	7.0	8.6

Table 1: The average user performance of accuracy and efficiency for 3 different pieces. *Events*: average number of identified note events per page. *FP*: average number of false positives per page. *FN*: average number of false negatives per page. *Time*: average time consumed per page.

Our "ground truth" is created by examining the symmetric difference between the two systems' note lists for a given page, resolving, by hand, each difference in favor of the correct interpretation. Of course it is possible that both systems make the same mistake, which would become part of the ground truth we use for evaluation. Thus we are really measuring the *difference* in note errors between the two systems, rather than an actual count of errors. From listening to the resulting "ground truth" we expect this discrepancy is insignificant.

We tested SmartScore with a single user while our system was tested with two different users. All the users were familiar with the system they were using and musically proficient enough to read the notation from original scores. In what follows we will call the SmartScore user **S** and the users of our system **A** and **B**.

We measure the accuracy of both systems using the error rate metric $r = (N_{fp} + N_{fn})/N_E$ where N_{fp} and N_{fn} are the total number of false positives and false negatives for one piece, and N_E is the total number of note events for that piece. As shown in Fig. 1, the results produced by the two systems were similar in terms of accuracy, both producing results in the 2-4% range after human correction.

Fig. 2 illustrates the efficiency of the two systems, measured in terms of keystrokes and mouse clicks (left) and clock time (right). In the Mozart and Chopin pieces there was comparable effort required between the two systems when measured in terms of user actions, though less clock time for our system. For the Ravel the clock time was about equal for the two systems, while our system used more keystrokes and mouse clicks. Overall the two systems' performance appears comparable. Table 1 presents the numeric results that summarize the experiments in terms of accuracy and effort.

[2] All the image data and symbolic output from both systems can be accessed from this website: https://sites.google.com/site/interactiveomr17/

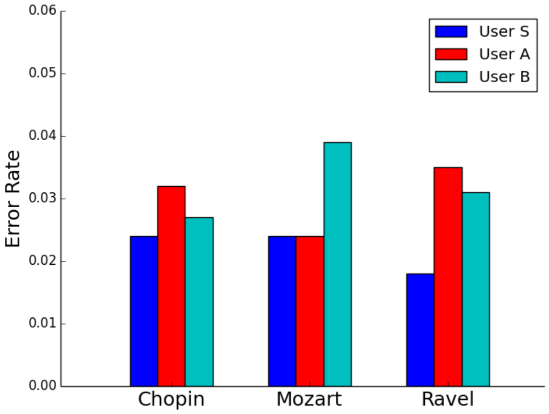

Figure 1: The error rate of recognized events after human proofreading and correction.

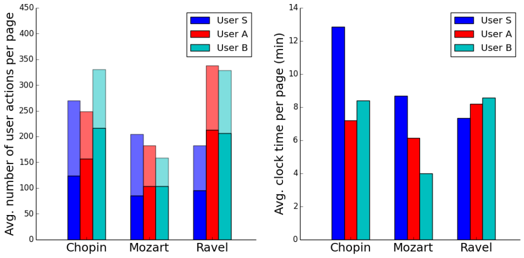

Figure 2: *Left*: average key strokes (bottom bar) and mouse clicks (top bar) used for each page in three different pieces. *Right*: average time consumed for each page in three different pieces.

In general, we conclude that our system demonstrates performance that is competitive with SmartScore, in terms of both accuracy and efficiency. The data suggest that SmartScore was slightly more accurate while our system was slightly faster — users make different tradeoffs between these two objectives. In addition, it is worth noting that the data produced by our system also preserves more information about the precise construction and location of image symbols. While beyond the scope of our evaluation, such information can be integral to renotation approaches that leverage specific layout information from the original when creating newly formatted music notation.

5 CONCLUSION

Our work with OMR demonstrates the value of human-guided systems in this domain. Even though the competing commercial system shows better raw recognition ability than ours, we were able to achieve comparable performance by casting the problem as one of human-in-the-loop computation. In doing so, we present a communication channel that allows the system to integrate the human input into the core recognition process. In contrast, the commercial system views this input as postprocess, thus missing an effective way to improve the system. This is the main "take away" message of the current effort. The challenges of OMR are familiar from a wide variety of different recognition problems, especially where the data are ambiguous, thus requiring human knowledge and experience to inform recognition. Thus we believe in the applicability of human-guided systems in a wide range of recognition problems far beyond the current focus on OMR.

REFERENCES

[1] P. Bellini, I. Bruno, and P. Nesi. 2007. Assessing Optical Music Recognition Tools. *Computer Music Journal* 31(1) (2007), 68–93.
[2] Taylor Berg-Kirkpatrick and Dan Klein. 2014. Improved Typesetting Models for Historical OCR. In *ACL (2).* 118–123.
[3] D. Blostein and H. S. Baird. 1992. A Critical Survey of Music Image Analysis. In *Structured Document Image Analysis.* 405–434.
[4] Steve Branson, Catherine Wah, Florian Schroff, Boris Babenko, Peter Welinder, Pietro Perona, and Serge Belongie. 2010. Visual recognition with humans in the loop. In *ECCV.* 438–451.
[5] John L. Bresina, Ari K. Jónsson, Paul H. Morris, and Kanna Rajan. 2005. Mixed-Initiative Activity Planning for Mars Rovers. In *IJCAI.* 1709–1710.
[6] Nicholas J Bryan, Gautham J Mysore, and Ge Wang. 2013. Source Separation of Polyphonic Music with Interactive User-Feedback on a Piano Roll Display. In *ISMIR.* 119–124.
[7] Liang Chen and Christopher Raphael. 2016. Human-Directed Optical Music Recognition. In *Document Recognition and Retrieval XXIII.* 1–9.
[8] Liang Chen, Erik Stolterman, and Christopher Raphael. 2016. Human-interactive optical music recognition. In *ISMIR.* 647–653.
[9] G. S. Choudhury, T. DiLauro, M. Droettboom, I. Fujinaga, B. Harrington, and K. MacMillan. 2000. Optical Music Recognition System within a Large-Scale Digitization Project. In *ISMIR.*
[10] Michael T Cox and Manuela M Veloso. 1997. Supporting combined human and machine planning: An interface for planning by analogical reasoning. In *International Conference on Case-Based Reasoning.* 531–540.
[11] George Ferguson and James F. Allen. 1998. TRIPS: An Integrated Intelligent Problem-Solving Assistant. In *AAAI 98, IAAI 98.* 567–572.
[12] I. Fujinaga. 1996. Exemplar-Based Learning in Adaptive Optical Music Recognition System. In *ICMC*, Vol. 12. 55–56.
[13] G. Jones, B. Ong, I. Bruno, and K. Ng. 2008. Optical Music Imaging: Music Document Digitisation, Recognition, Evaluation, and Restoration. *Interactive Multimedia Music Technologies* (2008), 50–79.
[14] Gary E Kopec, Philip A Chou, and David A Maltz. 1996. Markov source model for printed music decoding. *Journal of Electronic Imaging* 5, 1 (1996), 7–14.
[15] L. Pugin, J. A. Burgoyne, and I. Fujinaga. 2007. MAP Adaptation to Improve Optical Music Recognition of Early Music Documents Using Hidden Markov Models. In *ISMIR.* 513–516.
[16] A. Rebelo, G. Capela, and J. S. Cardoso. 2009. Optical Recognition of Music Symbols. *International Journal on Document Analysis and Recognition* 13 (2009), 19–31.
[17] Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Andre RS Marcal, Carlos Guedes, and Jaime S Cardoso. 2012. Optical music recognition: state-of-the-art and open issues. *International Journal of Multimedia Information Retrieval* 1, 3 (2012), 173–190.
[18] Verónica Romero, Alejandro H Toselli, Luis Rodríguez, and Enrique Vidal. 2007. Computer assisted transcription for ancient text images. In *Image Analysis and Recognition.* 1182–1193.
[19] F. Rossant and I. Bloch. 2007. Robust and Adaptive OMR System Including Fuzzy Modeling, Fusion of Musical Rules, and Possible Error Detection. *EURASIP Journal on Applied Signal Processing* (2007).
[20] SharpEye. 2008. http://www.music-scanning.com/sharpeye.html. (2008).
[21] SmartScore. 1991. http://www.musitek.com. (1991).
[22] Alexander Waibel, Rainer Stiefelhagen, Rolf Carlson, J Casas, Jan Kleindienst, Lori Lamel, Oswald Lanz, Djamel Mostefa, Maurizio Omologo, Fabio Pianesi, et al. 2010. Computers in the human interaction loop. In *Handbook of Ambient Intelligence and Smart Environments.* 1071–1116.