Optical Music Recognition for Interactive Score Display

Dan Ringwalt
Carnegie Mellon University
School of Computer Science
5000 Forbes Ave
Pittsburgh, PA
ringwalt@cmu.edu

Roger B. Dannenberg
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA
rbd@cs.cmu.edu

Andrew Russell
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA
andrewru@andrew.cmu.edu

ABSTRACT

Optical music recognition (OMR) is the task of recognizing images of musical scores. In this paper, we apply the first steps of optical music recognition, specifically staff, barline, and repeat detection, to automate annotation of scores for use in an interactive score display system. We developed new and simpler methods for these first steps, and show that our staff detection method outperforms existing methods on a previously used test set. Annotating staves and barlines by OMR and verifying by hand results in a 12-fold reduction in manual effort, facilitating the bulk import of a much larger collection of scores into our interactive score display system.

Author Keywords

Optical music recognition, Human-computer music performance, Live Score Display

ACM Classification

H.5.5 [Information Interfaces and Presentation] Sound and Music Computing—Methodologies and techniques,
I.7.5 [Document and Text Processing] Document Capture—Graphics recognition and interpretation.

1. INTRODUCTION

The computational power and storage capacity of modern computers (and even mobile computing devices) allows the construction of innovative musical interfaces and instruments that incorporate images of common practice music notation. Using images rather than structured machine-readable notation eliminates much of the difficulty of capturing and reusing score information that is already present in paper scores. Rather than painstakingly entering music into a music notation editor, one can simply scan or photograph existing printed music to obtain a digital image that can be displayed on a computer screen.

There are many applications of digital music displays: Performers can replace large folders of music with compact tablet computers for use on stage in live performance [18; 21]. Multi-media music players have been created that show the score synchronized with music playback [17]. Interactive music performances have employed music notation on digital displays to direct performers [10; 19]. Computer music accompaniment systems [7] sometimes use music displays to show music to the performer and automatically turn pages using score-following techniques [24; 25]. Digital music displays have been proposed as the basis for general-purpose interfaces to music sequencers and live music performance systems [8].

For scanned music notation images to be used effectively in interfaces, meta-data must be added to indicate the location of systems, barlines, and other musical information. By annotating some key features of the score image, one can enable automatic navigation, page-turning, extracting individual staves and systems to format them for small displays, and other interesting operations. Moreover, if one knows the beat position of each bar line in the displayed notation, one can use the notation as an index into the music: For example, pointing at a measure can indicate the start location for a music playback system.

Notice that labeled music images require more work than simply scanning images. Nevertheless, labeling staff and measure locations is a small fraction of the work of entering a full transcription of the score into a score editor. Alternatively, one might apply optical music recognition (OMR) followed by tedious corrections and manual formatting, but correcting every mistake, such as incorrect notes, is very time-consuming for most music [2]. An intermediate approach, which is the topic of this paper, is to use OMR *only for the required meta-data needed to navigate through a score.* In particular, automatically identifying staves and barlines alone can save almost all the work of preparing a scanned score for an interactive music display. The first steps of OMR to identify this meta-data are more accurate, and require much less time to correct. We introduce new OMR techniques and show that they result in an almost 12-fold speed-up in identifying staff systems, barlines, and repeats for use in music display applications.

In the following section, we describe some related work in score display systems and previous OMR systems. In Section 3, we describe new OMR techniques we have developed. In Section 4, we evaluate our system. In Section 5, we present conclusions and describe future work.

2. RELATED WORK

Our score display software has evolved from the Live Score Display (LSD) system by Zeyu Jin [15]. The Live Score Display system uses an *arrangement*, or mapping from time to a measure in the score [9], and therefore requires the score to be annotated with the locations of staves and barlines. An example of Live Score Display in typical use is shown in Figure 1.

With this program, the user first creates systems by clicking and dragging within the score images. The user can then click within each system to create barlines. If any systems or barlines are positioned poorly, the user can drag them to

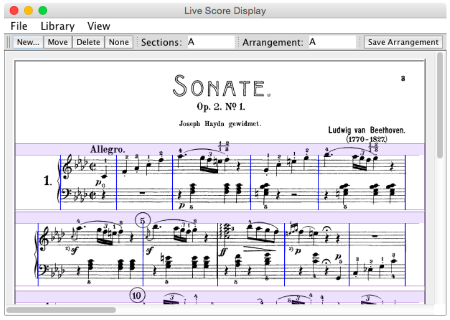

Figure 1: A typical score annotated in the Live Score Display system.

correct their position.

2.1 Staff Size Estimation

Musical staves may be characterized by two parameters. *staffthick* is the thickness of the staff lines, while *staffdist* is the sum of the thickness of the staff line and the space between staff lines (equivalently, the vertical distance between the center of two adjacent staff lines.)

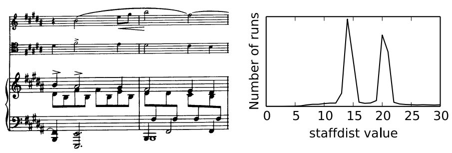

Figure 2: A section of a score (left) with the corresponding *staffdist* histogram (right). Each valid *staffdist* value has a prominent peak in the histogram.

Cardoso et. al's algorithm [5] was implemented for staff size estimation. Convert each column of the image to a run-length encoding, where runs are each consecutive sequence of the same pixel value. Compute a histogram of the sums of each 2 consecutive run lengths. Strong peaks correspond to valid values of *staffdist*; multiple values are possible when there is an *ossia* part or different instruments use different staff sizes. An example of the *staffdist* histogram is shown in Figure 2.

Next, given *staffdist*, we want to calculate *staffthick*. For each pair of runs which sum to *staffdist*, store the length of the black run. Make a histogram of the results, and set *staffthick* to be the peak of the histogram.

2.2 Rotation

A version of Lobb et al.'s windowed FFT method [18] is used to detect and reverse rotation of the page. Although the original method used grayscale images with a Hamming window, we found that a box window works well with binary images.

2.3 Previous Staff Detection Algorithms

The Gamera MusicStaves toolkit [6] contains 10 implementations of various staff detection and removal algorithms. Four of the best performing of these algorithms (Fujinaga's [14] and Dalitz's algorithms, "projections" and "skeleton" [6]) were chosen as a baseline to compare to a new algorithm. Capela et al.'s more recent Stable Paths algorithm [3] was implemented as another baseline.

Projections have been used extensively for optical music recognition [12], as they detect horizontal or vertical lines which appear in several contexts. Horizontal or vertical projections are defined as the sum of black pixels occurring in each row or column of the image, respectively. The most trivial staff detection algorithm simply searches for 5 equally spaced peaks in the horizontal projection of the image.

Bainbridge previously distinguished between algorithms which use horizontal projections to detect each staff line, or identify staff cross-sections in each column separately [2]. More generally, algorithms such as Stable Paths, which detect staff lines individually that are not necessarily horizontal, can be grouped into the first category.

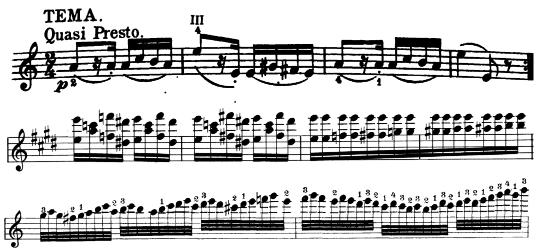

Figure 3: Various siuations which cause current staff detection algorithms to behave incorrectly. Top to bottom: "bowing" is possible with a tightly bound book, prominent ledger lines may be detected as part of a staff, and beams may occlude most of the staff.

Each category seems to behave poorly on some inputs, which are illustrated in Figure 3. The first category seems to be less robust to "bowing"—distortion caused by the page curving next to the spine [1]—or a rotated score; horizontal projections are known to work poorly in this case [2] and Stable Paths may detect staff lines that jump between actual lines [4]. On the other hand, the second category is vulnerable to detecting long ledger lines as part of a staff, as they form multiple possible staves in a cross-section. Additionally, much of the staff may be occluded by beams or other symbols, creating gaps where no staff cross-sections are detected.

3. NEW OMR TECHNIQUES

3.1 Staff Detection

A new staff detection algorithm, *Filtered Hough Transform*, was developed to solve these issues and simplify the overall staff detection process. We first noted that assuming a constant *staffdist* value, each possible staff in a column can be reduced to the y-coordinate of its center line. Next, all that is necessary is a simple line detection for each staff, and the other staff lines can be reconstructed by adding a multiple of *staffdist*. We have found that *staffdist* is constant for each page of a scanned score; Figure 2 is a representative example. Therefore, the reconstructed lines are accurate.

We implemented the first step as a filter operating on each pixel. We set the pixel to white if any pixels 1 or 2∗*staffdist* above or below are white. Finally, we noted that even when the staff is obscured, at least 2 lines are usually visible. Therefore, at least 2 of the possible staff lines (the original point and the 4 points as described above) must have a white pixel *staffthick* above and below. The result of the filter is shown in Figure 4.

Next, the Hough transform [11] is used to detect lines in the filtered image. In the context of OMR, the Hough transform is equivalent to a 2D array containing the hori-

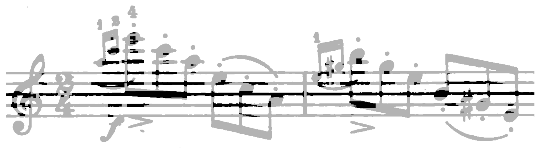

Figure 4: The results of the staff center filter (black) on the original image (gray).

zontal projection of several rotated versions of the image. Although the Hough transform has previously been used for staff detection [20], it obtains poor results similar to horizontal projection for bowed scores [22]. However, the staff center filter leaves the center line more prominent than the others, so it is still easily detected even if the other lines bleed together due to bowing.

To avoid ledger lines being detected, we want to only choose the largest peak in parameter space near each staff. Therefore, after the peak is found in the accumulator array, all lines with an intercept within 6∗*staffdist* are set to 0.

3.2 Barline and System Detection

Compared to staff detection, fewer methods have been described which detect barlines and join staves into staff systems. Barlines are commonly identified as vertical lines which span the height of the staff and have no adjacent noteheads [13; 23]. Since our system does not yet classify notes or other symbols, we needed to develop a different heuristic for detecting barlines.

Without classifying notes, we extract each staff from the image and take the vertical projection (the number of black pixels in each column.) Barlines are classified as consecutive columns with at least 75% black pixels, surrounded by columns with at most 10% black pixels.

Afterwards, we detect systems as follows. If consecutive staves all share a barline in the same position, then we fit a line through the barlines. If there are black pixels along the line for any shared barline, allowing for only small gaps, the staves are joined into one system.

3.3 Repeat Detection

Repeat dots must appear next to a normal barline, which in turn is next to a thick barline. Previous work has searched for repeat dots any time two barlines are close together [13]. As two close barlines may be detected as one barline in the projection process, we only look for thick barlines, and then merge any close barlines together. If we found a thick barline, we perform connected component analysis on the staff and identify two small dots on one side of the thick barline as a repeat symbol. Checking a smaller number of staves which contain a thick barline also improves processing time, as our implementation of connected component analysis seems to take longer than the other steps.

4. EVALUATION

4.1 Staff Detection

We used the Gamera MusicStaves test set [6] to evaluate the selected baseline algorithms and the Filtered Hough Transform. The test set uses scores from music engraving software with random distortions applied, where the actual staff locations are known. Most distortions used Kanungo noise with several different parameters, as this emulates local noise caused by the scanning process [16]. The “curvature”, “rotation”, and “white_speckles” deformations were

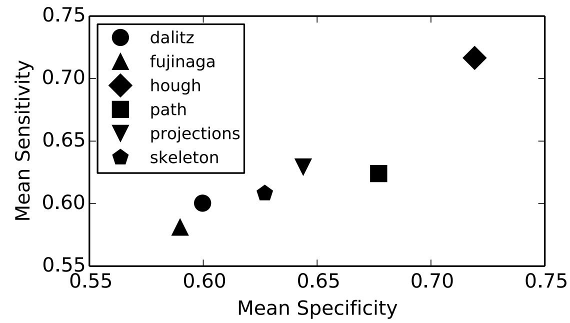

Figure 5: Using engraved images from the Gamera MusicStaves test set, Filtered Hough Transform outperforms the baseline algorithms toolkit on both sensitivity and specificity.

	Sensitivity	Specificity
Staff Systems	100 %	100 %
Barlines	99.8 %	100 %
Repeats	87.5 %	87.5 %

Table 1: Performance on the OMR steps occurring after staff detection.

also used to emulate other types of noise. Although the original tests measured the accuracy of a staff removal step, staff removal was not useful for our application, and we only implemented a naive staff removal method in our system. Therefore, we simply checked whether each detected staff was within *staffdist/2* of a known staff, and evaluated sensitivity and specificity. The mean sensitivity and specificity of each algorithm is shown in Figure 5. Our method (*hough*) demonstrated a 9% increase in sensitivity and 4% increase in specificity compared to the other algorithms.

4.2 Staff Systems, Barlines, and Repeats

Sensitivity and specificity on the later OMR steps were measured by hand on a 17-page scan of Ludwig van Beethoven’s *Piano Sonata, Op. 2, No. 1*, which was obtained from IMSLP. The results are presented in Table 1. All staves and staff systems were correctly detected.

4.3 Annotation Time

The same score was used to measure the time required for OMR-assisted and manual score annotation. First, the score was labelled fully by hand. Next, the score was processed by OMR with the results imported into Live Score Display, and then corrected by hand. We found an almost 12-fold speedup in the manual work (Table 2).

5. CONCLUSIONS

We developed new methods for extracting structural information from a score without performing full OMR, allow-

	Processing	Manual work	Total
Manual	N/A	18 m 24.1 s	18 m 24.1 s
OMR	23.0 s	1 m 36.0 s	1 m 59.0 s
Speedup	N/A	11.5 X	9.3 X

Table 2: Time spent correcting OMR output vs. annotating the score by hand.

ing scanned scores to be added to a score display system more quickly. Our staff detection method, Filtered Hough Transform, combines the advantages of several existing staff detection algorithms, while it is even simpler. The first staff filter step is more relaxed than other methods which operate on columns of the image, so it does not filter out partially occluded staves. The second Hough transform step finds a globally optimal best fit line, unlike existing methods that only join adjacent columns of the image, increasing robustness to inaccuracy in small sections.

We found that on a synthetic data set, our method outperformed the baseline staff detection algorithms. Real scans may contain more global deformation due to page curl. Greater robustness may be added using a preprocessing step such as Fujinaga's deskewing [13].

Our system has also validated the use of a single staff center line, combined with the value of *staffdist*, as opposed to building 5 separate staff lines. Removing unnecessary parameters in our staff model should increase the robustness of staff detection and simplify later steps in our system.

We also implemented barline, system, and repeat detection, without a complete OMR system. We showed that using simpler versions of these steps, without relying on other information such as note positions, worked robustly.

Finally, we demonstrated that our system reduces the manual work required for score annotation by 12-fold. This will help prevent repetitive strain injury from manually annotating scores, and allow many more scores to be used in the interactive score display system.

References

[1] D. Bainbridge. *Extensible optical music recognition*. PhD thesis, University of Canterbury, 1997.

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

[3] A. Capela, A. Rebelo, J. S. Cardoso, and C. Guedes. Staff line detection and removal with stable paths. In *SIGMAP 2008 - Proceedings of the International Conference on Signal Processing and Multimedia Applications*, pages 263–270, 2008.

[4] J. S. Cardoso, A. Capela, A. Rebelo, and C. Guedes. A connected path approach for staff detection on a music score. In *15th IEEE International Conference on Image Processing*, pages 1005–1008. IEEE, 2008.

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

[6] 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):753–66, May 2008.

[7] R. B. Dannenberg. An on-line algorithm for real-time accompaniment. In *Proceedings of the 1984 International Computer Music Conference*, pages 193–198. Computer Music Association, 1985.

[8] R. B. Dannenberg, N. E. Gold, D. Liang, and G. Xia. Active Scores: Representation and synchronization in human-computer performance of popular music. *Computer Music Journal*, 38(2):51–62, 2014.

[9] R. B. Dannenberg and A. Russell. Arrangements: Flexibly Adapting Music Data for Live Performance. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME 2015)*, 2015.

[10] N. Didkovsky. Recent compositions and performance instruments realized in the Java music specification language. In *Proceedings of the 2004 International Computer Music Conference*, pages 746–749, 2004.

[11] R. O. Duda and P. E. Hart. Use of the Hough transform to detect lines. Technical Report April 1971.

[12] I. Fujinaga. Optical music recognition using projections. Master's thesis, McGill University, 1988.

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

[14] I. Fujinaga. Staff detection and removal. *Visual perception of music notation: on-line and off-line recognition*, pages 1–39, 2004.

[15] Z. Jin and R. B. Dannenberg. Formal semantics for music control flow. In *Proceedings of the 2013 International Computer Music Conference*, pages 85–92, 2013.

[16] 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. IEEE, 1993.

[17] F. Kurth, M. Meinard, D. Damm, C. Fremerey, A. Ribbrock, and M. Clausen. SyncPlayer – An advanced system for multimodal music access. In *Proceedings of the 8th International Conference on Music Information Retrieval*, pages 381–388. ISMIR, 2007.

[18] R. Lobb, T. Bell, and D. Bainbridge. Fast capture of sheet music for an agile digital music library. In *Proceedings of the 6th International Conference on Music Information Retrieval*, pages 145–152. ISMIR, 2005.

[19] C. McClelland and M. Alcorn. Exploring new composer/performer interactions using real-time notation. In *Proceedings of the International Computer Music Conference*, pages 176–179, 2008.

[20] H. Miyao, T. Ejima, M. Mihayara, and K. Kotani. Symbol extraction for printed piano scores based on musical knowledge. *IEICE Transactions*, J75-D-II(11):1848–1855, 1992.

[21] MusicNotes. "Musicnotes.com." Available: www.musicnotes.com. [Accessed: 29-Jan-2015].

[22] R. Randriamahefa, J. P. Cocquerez, C. Fluhr, F. Pépin, and S. Philipp. Printed music recognition. In *Proceedings of the Second International Conference on Document Analysis and Recognition*, pages 898–901. IEEE, 1993.

[23] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. S. Marcal, C. Guedes, and J. S. Cardoso. Optical music recognition: state-of-the-art and open issues. *International Journal of Multimedia Information Retrieval*, 1(3):173–190, March 2012.

[24] SmartMusic. "SmartMusic." Available: www.smartmusic.com. [Accessed: 29-Jan-2015].

[25] Tonara. "Tonara." Available: www.tonara.com. [Accessed: 29-Jan-2015].