OPTICAL MEASURE RECOGNITION IN COMMON MUSIC NOTATION

Gabriel Vigliensoni, Gregory Burlet, and Ichiro Fujinaga
Centre for Interdisciplinary Research in Music Media and Technology (CIRMMT)
McGill University, Montréal, Québec, Canada
{gabriel,ich}@music.mcgill.ca, gregory.burlet@mail.mcgill.ca

ABSTRACT

This paper presents work on the automatic recognition of measures in common Western music notation scores using optical music recognition techniques. It is important to extract the bounding boxes of measures within a music score to facilitate some methods of multimodal navigation of music catalogues. We present an image processing algorithm that extracts the position of barlines on an input music score in order to deduce the number and position of measures on the page. An open-source implementation of this algorithm is made publicly available. In addition, we have created a ground-truth dataset of 100 images of music scores with manually annotated measures. We conducted several experiments using different combinations of values for two critical parameters to evaluate our measure recognition algorithm. Our algorithm obtained an f-score of 91 percent with the optimal set of parameters. Although our implementation obtained results similar to previous approaches, the scope and size of the evaluation dataset is significantly larger.

1. INTRODUCTION

Optical music recognition (OMR) is the process of converting scanned images of pages of music into computer readable and manipulable symbols using a variety of image processing techniques. Thus, OMR is seen as a valuable tool that helps accelerate the creation of large collections of searchable music books.

However, the automatic recognition of printed music presents several substantial challenges, including: A large variability in the quality of analog or digital sources; the common superimposition of shapes within a score on staves, making it difficult for computers to isolate musical elements and extract musical features that represent the content; and finally, a large number of music symbols that can create a large pattern space [5]. Also, as noted in [7], a common source of OMR errors originate from the misinterpretation of note stems or other vertical structures in the score as barlines, or vice-versa, which leads to measure annotations with false positives or false negatives.

In this project we aim to create an optical measure recognition algorithm capable of recognizing the physical location of barlines in a wide range of scores of common Western music notation (CWMN), deriving the bounding boxes for the measures, and storing these elements in a symbolic music file format. This allows us to relate the physical location on the page to the structure of the music itself. Applications that employ optical measure recognition can enhance interactions with music scores, and include multimodal music presentation and navigation, such as synchronizing digitized scores with audio playback [8], or content-based retrieval systems, which allow users to query the score by its measures [9]. In these applications, the correct extraction of barlines is essential for a proper alignment of the different music representations.

The structure of this paper is as follows: Section 2 presents the structure and implementation details of the optical measure recognition algorithm we have developed. Section 3 describes the annotation methodology and design of the ground-truth dataset that is used to evaluate the optical measure recognition algorithm. The evaluation procedure is presented in Section 4. We conclude in Section 5 with a discussion of our results and future work.

2. MEASURE RECOGNITION ALGORITHM

Our technique for locating the bounding boxes of measures within a music score relies on several image processing functions and follows the task model proposed by Bainbridge and Bell [1], which decomposes the problem of OMR into several key stages: image preprocessing and normalization, staffline identification and removal, musical object location, and musical reasoning.

After preprocessing the input image, stafflines are removed from the score and thin, long, vertical lines that have the same horizontal position within a system of music are located, even if they are connected. The connected height of these elements should approximately equal the height of the system to which they belong. Our approach assumes that barlines are usually taller than the stem of notes.

We experimented with several image processing algorithms in our optical measure recognition system. The Gamera software framework [10] provides a flexible and extensible environment for testing these methods and implementing new ones, if desired. Fig. 1 displays intermediary output of our measure recognition system at different

stages of processing, described in the following sections, on a portion of an image selected from our dataset.

2.1 Preprocessing

Before analyzing the input music score, the image must undergo two preprocessing steps: binarization and rotation correction. The binarization step coerces each pixel in the image to be either black or white according to a specified threshold parameter and is accomplished by using the Otsu binarization algorithm [12]. The rotation correction step automatically rotates skewed images and is accomplished by using the `correct_rotation` method that is part of the document-preprocessing bundle toolkit for Gamera [13]. [1]

2.2 Staff grouping hint

Our measure recognition algorithm requires prerequisite information, supplied by humans, that describes the structure of the staves on the music score being processed. This information, hereinafter referred to as the *staff grouping hint*, indicates how many staves are on the page, how many systems are on the page, how staves are linked together into systems, and whether barlines span the space between groups of staves.

The staff grouping hint is a string that encodes the structure of staves. For example, consider a page of piano music consisting of two staves that are broken into five systems, where barlines span the space between the two staves in each system, as in Fig. 2. The appropriate staff grouping hint for this page is `(2|)x5`, where parentheses indicate a group of staves, the pipe character `|` denotes that barlines span the space between staves in the group, and the `x` character indicates the number of systems the staff group is broken into. [2]

Although the staff group hint can be seen as a bottleneck because it requires human intervention, it is an important component of our system because it is used to properly encode the output symbolic music file and for fault detection of the staff detection algorithm, described in the next section. Also, most multi-page scores do not change their system structure across pages, and so a hint created for one page can often be used for the whole score.

2.3 Staff detection and removal

After preprocessing the input image, a staff detection algorithm searches for staves on the music score and returns the bounding box information for each staff. A staffline removal algorithm then discards the located stafflines from the music score. However, staff detection and removal algorithms yield variable results depending on the notation style of the music score, image scan quality, and the amount of noise (artifacts) present in the image. As a result of the high variability of images in our dataset, we could not rely on only one approach for detecting stafflines.

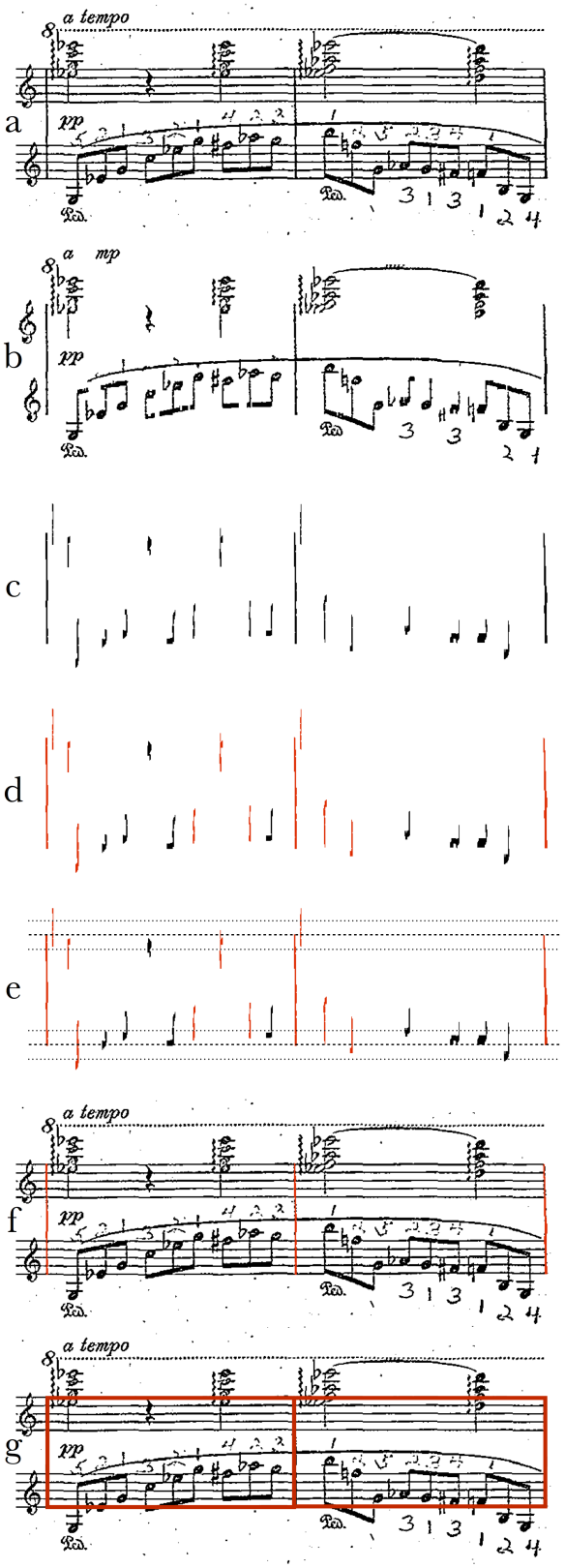

Figure 1. Process of extracting bar candidates. a) Original image, b) stafflines removed on preprocessed image, c) most frequent vertical black runs removed, d) barline candidates filtered by aspect ratio, e) filtering of bar candidates by vertical tolerance thresholding, f) final barlines, and g) final retrieved measures superimposed on the original image.

[1] `https://github.com/DDMAL/document-preprocessing-toolkit`

[2] More staff grouping hint examples can be accessed at `http://ddmal.music.mcgill.ca/optical_measure_recognition_staffgroup_hint_examples`

Low-quality scans and images with an excessive amount of artifacts frequently cause staff detection algorithms to fail, and so we implemented an error handling approach that tries one algorithm first, and if that fails, an alternative is used instead. We consider a staff detection algorithm to have failed when the number of detected staves does not equal the value derived from the provided staff grouping hint.

Following previous research [15], we used the *MusicStaves* Gamera Toolkit because it offers a number of different algorithms for detecting the position of stafflines in an image, and also for removing them. [3] We used the *Miyao* and *Dalitz* algorithms to perform staff detection. The *Miyao* algorithm provides a precise method for determining horizontal staff slope changes in inclined, broken, or non-linear staves by breaking the staffline into equidistant segments and capturing the vertical position of each segment [11]. The *Dalitz* staff detection algorithm [3] is also capable of tracking non-straight lines by using long quasi-black run extraction and skelenotization (i.e., the representation of each staffline as a one point thick continuous path), but it does not break the stafflines into equidistant segments. Our approach for finding measures depends heavily on detecting the position of the staff, and so we implemented two approaches in case one of them fails. We tested both configurations (i.e., *Dalitz-Miyao* and *Miyao-Dalitz*), and concluded that the former arrangement yields superior performance with respect to our ground-truth dataset. After successful recognition of the position of staves, the bounding box of each system can be calculated using the provided staff grouping hint.

Following the staff detection stage, staffline removal must be performed to eliminate each staffline within the music score. This process is important because it isolates superimposed music symbols on staves, facilitating their recognition. However, a comparative study established that there is no single superior algorithm for performing staffline removal [4]. Using a number of different metrics on images with deformations, we observed that the performance of many algorithms for staffline removal is similar, with no one technique being obviously better in general. Based on our previous work [15], we chose the Roach & Tatem staffline removal algorithm [14] in our optical measure recognition system. Fig. 1(b) shows the output of the staffline removal algorithm on a portion of a preprocessed image from our dataset.

2.4 Locating barline candidates

Once the position of each staff and system is calculated and all stafflines have been removed, we filter short vertical runs of black pixels in order to remove ligatures, beams, and other elements on the page that are unlikely to be candidates for a barline. The most frequent run-length is calculated and is used in subsequent processing steps. Since removing stafflines and short vertical runs frequently leaves unwanted artifacts on the page, we finally despeckle the image to remove all connected components smaller than a threshold value, which is dependent on the most frequent run-length value.

Once we have removed the horizontal lines from the image, we perform a connected components analysis to segment all of the residual glyphs on the page. The resulting set of connected components is filtered to only include thin, vertical elements, which are referred to as *bar candidates*. The discriminating feature for the selection of bar candidates is the *aspect ratio*: the relation between the width and the height of a component. Fig. 1(c) shows the result of filtering short vertical runs and despeckling the image. Fig. 1(d) highlights bar candidates that have an acceptable aspect ratio.

A bar candidate may be broken into several unconnected lines, depending on the quality of the original image, the effects of any of the intermediary processing steps, or simply from barlines that intentionally do not span an entire system. If bar candidates within a system have roughly the same horizontal position, they are connected into a single bar candidate. The height of each connected bar candidate is calculated and compared to the height of the system to which it belongs; these heights should approximately be the same. Moreover, the upper and lower vertical position of the bar candidate should lie sufficiently close to the upper and lower vertical position of its system, respectively. If the bar candidate fails to meet this criterion, the glyph is discarded. The sensitivity of this filtering step is controlled by the *vertical tolerance* parameter. Fig. 1(e) shows a visual representation of the vertical tolerance filtering process.

An additional filtering step addresses two common cases whereby certain bar candidates are included as false positives: The first situation occurs when accidentals preface the musical content on a staff. As most horizontal lines are removed in previous processing steps, the vertical lines that remain in the accidentals are usually horizontally aligned across the staves. Therefore, they are linked together into a single bar candidate, which results in a false positive. The second situation occurs when a double barline is considered as two bar candidates. Considering that the largest key signature has seven accidentals spanning twice the vertical size of the staff, we resolve both of the aforementioned issues by filtering bar candidates that are less than twice the height of the staff apart in the horizontal direction.

2.5 Encoding the position of measures

The result of the filtering processes is a set of recognized barline candidates from an input page of music, as seen in Fig. 1(f). These barlines candidates are sorted according to their system number and their horizontal position within the system. The bounding box for each measure is calculated by considering the location and dimensions of sequential barlines in each system, as seen in Fig. 1(g). The resulting set of measure bounding boxes is encoded in the Music Encoding Initiative (MEI) file format. [4] The MEI file format is used because of its ability to record the

[3] http://lionel.kr.hs-niederrhein.de/~dalitz/data/projekte/staffline

[4] http://music-encoding.org

structure of music entities as well as the physical position of all elements on the page [6]. Also encoded in this file is the overall structure of the score that indicates which measures belong to which system and which staves belong to which system.

3. GROUND-TRUTH DATASET

To the best of our knowledge, there are no standard OMR datasets that are complete with annotated measure bounding boxes. Therefore, we created our own to test and evaluate the performance of our optical measure recognition system. Our dataset consists of 100 pages extracted from the International Music Score Library Project (IMSLP). [5] We chose to extract images from IMSLP because of the quantity and diversity of the CWMN materials in its database.

To create this dataset we selected a random musical work from IMSLP, downloaded a random score or part from this work, and finally, selected a page at random from the score or part. As the purpose of our study is to locate the position of measures on pages of music, images of blank pages, pages with mostly text, and pages with no measures were manually discarded and replaced. In the initial draw of the dataset, images with these characteristics accounted for roughly 15 percent of the dataset. The set of downloaded images were in portable document format (PDF), which were processed using the libraries *pyPDF* [6] and *pdfrw*, [7] and converted to the tagged image file format (TIFF) using the *Adobe Acrobat Professional* application.

3.1 Measure annotations

Once the images in the dataset were transformed into the desired TIFF format, we created a ground-truth dataset of manually annotated bounding boxes for all measures on each page of music. We developed a Python application to perform the annotations. [8] The graphical user interface of the application displays an image to be annotated by a user, who indicates the presence of a measure on the page by clicking on the top-left position of a measure and dragging the mouse to the bottom-right corner of the measure. In order to ensure that all stafflines are straight, the image displayed to the annotators was automatically rotated using the same algorithm as in the preprocessing step of the presented optical measure recognition algorithm. The application encodes and saves the annotations as an MEI file, using a similar structure as the output of the optical measure recognition algorithm.

Two annotators with musical training of at least 10 years were hired to annotate the bounding box of each measure occurring in the entire dataset, as well as to provide a text file containing the staff grouping hints for each image. The annotators were instructed to track the time and number of pages they annotated per session, and to start annotating at opposite ends of the dataset to reduce the chances of error

[5] http://imslp.org
[6] http://pybrary.net/pyPdf
[7] http://code.google.com/p/pdfrw
[8] https://github.com/DDMAL/barlineGroundTruth

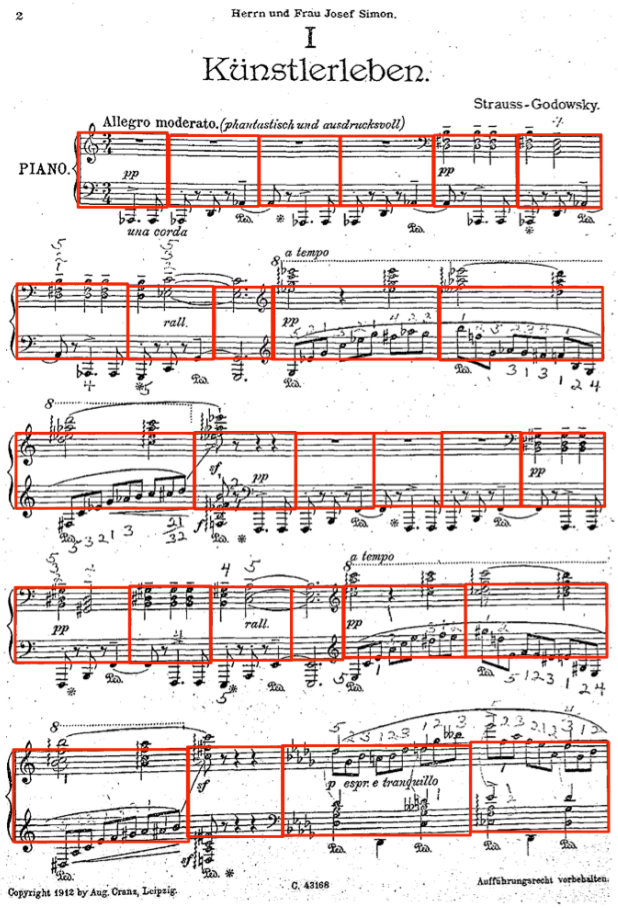

Figure 2. Manual measure annotations created using our standalone Python application for the image *IMSLP08436*, extracted from the International Music Score Library Project.

in the initial pages of the dataset. On average the annotators required 10 minutes to annotate the measures and create the staff group hint for each page. There were few discrepancies between the two annotators; the most common inconsistency was the staff group hint for complex pages of music. The dataset consists of 2,320 annotated measures, with a mean of μ = 23.43 measures per page, and a standard deviation of σ = 21.34. Fig. 2 displays a page of music from our dataset with the measure annotations superimposed.

Even with trained annotators, we encountered several challenges in the creation of this ground-truth dataset. Several recurrent issues arose during the annotation process, including how to interpret measures that are interrupted by a system break, how to annotate anacruses ("pick-up" notes), how to annotate repetition measures, and how to annotate measures that indicate changes in key signature but contain no notes. As our approach for finding measures on the score relies only on visual cues, all of the aforementioned cases are interpreted as separate measures. As such, we considered a measure interrupted by a system break, as well as an anacrusis, as two different measures. In addition, repeated measures were considered a single measure. Similar projects that recognize regions of a music score

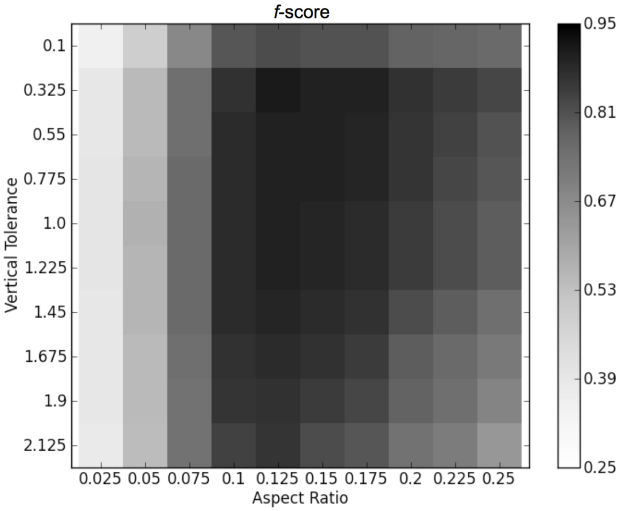

Figure 3. f-score results of the optical measure recognition algorithm. The x-axis displays different values of the *aspect ratio threshold* parameter. The y-axis displays different values of the *vertical tolerance threshold* parameter.

and synchronize these to audio playback have taken a similar approach, not yet considering repetitions of particular parts of a musical work [9]. Finally, areas of systems that contained the clef and accidentals, but no notes, were also considered to be a measure.

4. ALGORITHM EVALUATION

The performance of our optical measure recognition algorithm was evaluated by computing precision, recall, and f-score statistics for the automatically recognized measures on each page of music in the ground-truth dataset. Since we are not only concerned with the number of retrieved measures but also their size and physical position on the page, a measure is considered correctly recognized if its bounding box coordinates are within a quarter of an inch of its corresponding bounding box in the ground-truth measure annotations. [9]

Experiments were conducted to investigate the impact of different values of the two critical parameters of our optical measure recognition algorithm, namely the *aspect ratio* and the *vertical tolerance* parameters, described in Section 2. We iterated over a set of 100 combinations of these parameters and reported the resulting precision, recall, and f-score of our algorithm in each case.

Fig. 3 presents the results of the experiments across the entire dataset. It can be seen that the f-score value was highly influenced by the aspect ratio parameter. When this parameter was < 0.1 the f-score of our algorithm significantly decreased. This finding is intuitive because the appearance of elements on a music score are often variable, especially if it is handwritten. Consequently, it is unlikely to encounter barlines that are perfectly vertical with a small, constant width; they are typically skewed due to deformations in the image scan or contain artifacts resulting from intermediary processing steps of the optical music recognition algorithm.

The vertical tolerance threshold parameter, on the other hand, was found to not significantly affect the performance of the algorithm, especially when the aspect ratio threshold parameter was set to an optimal value. Only with extremely low values of vertical tolerance (i.e., when the tolerance was so small that the height of a bar candidate was expected to be almost the same as the system's height) did this parameter decrease the performance of the system. High values of this parameter also decreased the performance, but to a lesser degree.

Overall, the aspect ratio parameter had the most impact on the performance of the algorithm, though, both parameters exhibited a range of optimal values: aspect ratio threshold $\in [0.125, 0.150]$ and vertical tolerance threshold $\in [0.325, 1.00]$, which yielded an average f-score of 0.91 across the entire dataset. Similar barline recognition results have been obtained by commercial OMR systems [2]; however, in that study the evaluation dataset consisted of only five images and the algorithms being evaluated were undisclosed. Furthermore, Fotinea et al. [5] reported similar results on a dataset containing two pages of music.

Finally, certain pages in the dataset failed with all combinations of parameter values. The quality of these pages were generally quite poor and had discontinuous stafflines, which caused the staff detection algorithms to fail. Nevertheless, these pages were still included in the algorithm evaluation and resulted in an f-score of zero. We believe this accurately reflects how our system would perform in a "real-world" scenario.

5. CONCLUSION

We have presented work on developing a system that performs optical measure recognition on CWMN scores. Our approach follows an OMR workflow that includes image preprocessing, staff removal, and musical glyph recognition. Once all stafflines and short vertical runs of black pixels are removed, the algorithm finds thin, vertical elements on the page to form a set of barline candidates. Several heuristics were employed to filter this set of barline candidates into a final set of barlines, which were then used to calculate the bounding boxes of measures on the page. Our algorithm solely identifies measures on images of music scores, and thus, does not recognize other musical symbols such as the repeat sign, which instructs the performer to repeat a measure of music. This is problematic for applications that intend to synchronize digitized scores with audio playback and is an issue to address in future versions of our measure recognition system.

In order to test and evaluate our system, we manually annotated measure positions in 100 random pages of music from IMSLP and compared the bounding boxes produced by our optical measure recognition algorithm to the manual annotations using several descriptive statistics. We conducted several experiments to test different combinations

[9] Measurements in inches are converted to pixels using the *pixels per inch* parameter from the metadata of each image in the dataset.

of two critical parameters of our algorithm and discovered that the aspect ratio of a glyph is the most important discriminating feature for barlines. With optimal parameters, our algorithm obtained 91 percent f-score across the entire dataset.

Although our approach obtained similar results as previous systems, the scope and size of our evaluation dataset is much larger than those in the literature. We hope that the open-source, command line-based implementation of our system [10] can be easily integrated into existing OMR systems, and will stimulate future work in this area and help other researchers discover new ways to extract meaningful information from images of music scores.

6. ACKNOWLEDGEMENTS

The authors would like to thank our great development team for their hard work: Nick Esterer, Wei Gao, and Xia Song. Special thanks also to Alastair Porter for his invaluable insights at the beginning of the project. This project has been funded with the generous financial support of the *Deutsche Forschungsgemeinschaft* (DFG) as part of the Edirom project, and the *Social Sciences and Humanities Research Council* (SSHRC) of Canada.

7. REFERENCES

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

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

[3] Dalitz, C., T. Karsten, and F. Pose. 2005. Staff Line Removal Toolkit for Gamera. `http://music-staves.sourceforge.net`

[4] Dalitz, C., M. Droettboom, B. Pranzas, and I. Fujinaga. 2008. A comparitive study of staff removal algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30 (5): 753–66.

[5] Fotinea, S., G. Giakoupis, A. Liveris, S. Bakamidis, and G. Carayannis. 2000. An optical notation recognition system for printed music based on template matching and high level reasoning. In *Proceedings of the International Computer-assisted Information Retrieval Conference*, Paris, France, 1006–14.

[6] Hankinson, A., L. Pugin, and I. Fujinaga. 2010. An interchange format for optical music recognition applications. In *Proceedings of the International Society for Music Information Retrieval Conference*, Utrecht, Netherlands, 51–6.

[10] `https://github.com/DDMAL/barlineFinder`

[7] Knopke, I., and D. Byrd. 2007. Towards Musicdiff: A foundation for improved optical music recognition using multiple recognizers. In *Proceedings of the International Society for Music Information Retrieval Conference*, Vienna, Austria, 123–6.

[8] Kurth, F., M. Müller, C. Fremerey, Y. Chang, and M. Clausen. 2007. Automated synchronization of scanned sheet music with audio recordings. In *Proceedings of the International Society for Music Information Retrieval Conference*, Vienna, Austria, 261–6.

[9] Kurth, F., D. Damm, C. Fremerey, M. Müller, and M. Clausen. 2008. A framework for managing multimodal digitized music collections. In *Research and Advanced Technology for Digital Libraries. Lecture Notes in Computer Science* 5173: 334–45. Springer Berlin Heidelberg.

[10] MacMillan, K., M. Droettboom, and I. Fujinaga. 2002. Gamera: Optical music recognition in a new shell. In *Proceedings of the International Computer Music Conference*, La Habana, Cuba, 482–5.

[11] Miyao, H., and M. Okamoto. 2004. Stave extraction for printed music scores using DP matching. *Journal of Advanced Computational Intelligence and Intelligent Informatics* 8 (2): 208–15.

[12] Otsu, N. 1979. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics* 9 (1): 62–6.

[13] Ouyang, Y., J. Burgoyne, L. Pugin, and I. Fujinaga. 2009. A robust border dectection algorithm with applications to medieval music manuscripts. In *Proceedings of the International Computer Music Conference*, Montréal, Canada, 101–4.

[14] Roach, J., and J. Tatem. 1988. Using domain knowledge in low-level visual processing to interpret handwritten music: An experiment. *Pattern Recognition* 21 (1): 33–44.

[15] Vigliensoni, G., J. A. Burgoyne, A. Hankinson, and I. Fujinaga. 2011. Automatic pitch detection in printed square notation. In *Proceedings of the International Society for Music Information Retrieval Conference*, Miami, FL, 423–8.