LYRIC EXTRACTION AND RECOGNITION ON DIGITAL IMAGES OF EARLY MUSIC SOURCES

John Ashley Burgoyne **Yue Ouyang** **Tristan Himmelman**
Johanna Devaney **Laurent Pugin** **Ichiro Fujinaga**

Centre for Interdisciplinary Research in Music and Media Technology
McGill University
Montréal, Québec, Canada
{ashley,devaney,laurent,ich}@music.mcgill.ca
{yue.ouyang,tristan.himmelman}@mail.mcgill.ca

ABSTRACT

Optical music recognition (OMR) is one of the most promising tools for generating large-scale, distributable libraries of musical data. Much OMR work has focussed on instrumental music, avoiding a special challenge vocal music poses for OMR: lyric recognition. Lyrics complicate the page layout, making it more difficult to identify the regions of the page that carry musical notation. Furthermore, users expect a complete OMR process for vocal music to include recognition of the lyrics, reunification of syllables when they have been separated, and alignment of these lyrics with the recognised music. Unusual layouts and inconsistent practises for syllabification, however, make lyric recognition more challenging than traditional optical character recognition (OCR). This paper surveys historical approaches to lyric recognition, outlines open challenges, and presents a new approach to extracting text lines in medieval manuscripts, one of the frontiers of OMR research today.

1. INTRODUCTION

Researchers in music information retrieval (MIR) have gradually been building bigger databases of music that will enable large-scale computational musicology. One tool to expedite the development of such databases for older music is optical music recognition (OMR), the musical analogue to optical character recognition (OCR). When developing databases of vocal music, however, OMR alone is not enough: lyrics need to be recognised and stored along with the music, and because of certain particularities of musical notation, standard OCR tools are often insufficient for this task. Moreover, lyrics complicate the page layout, making it more difficult to conduct the basic image processing necessary to feed the OMR and OCR pipelines

Vocal music predominates among medieval music manuscripts, copied by hand from the ninth through the sixteenth centuries, as well as early printed music from the fifteenth and sixteenth centuries. These sources pose still more challenges for OMR and lyric recognition. Due to the relatively loose document production techniques of the time, page layouts can be highly non-standard and oriented more for display than consumption. Furthermore, as a result of ageing, early music documents are often in physically poor condition. Typical problems include non-uniform illumination, stains, and irregular page shape. Ink frequently has bled through from the reverse side of the page or shows through as a result of high-contrast microfilm photography. All of these degradations can inhibit the performance of segmentation (identifying the regions of the image that correspond to musical notation, lyrics, or other elements) and recognition (interpreting image shapes as musical notes or letters) [1]. Finally, scanning conventions for early documents themselves can require extra preprocessing to remove elements like rulers and colour bars before images are even sent to an OMR system [2].

This paper outlines some previous approaches to lyric recognition, highlights the open challenges, especially with respect to early documents, and presents two tools we have developed to facilitate work with digital images of early music: a new lyric editor for the Aruspix OMR package [3] and a new technique for extracting lyric lines from digital facsimiles of medieval manuscripts.

2. BACKGROUND

OMR systems have been working to handle lyrics for some time, and most systems share common challenges. One is layout analysis: how can these systems determine which regions of a page carry music and which lyrics? Some systems rely on heuristics based on projections, run lengths, and other local image features, generally seeking to extract the music first and define the remaining regions as text [3,4]; others run OCR first, using those regions where the system successfully identifies letters as lyric regions [6]. Once the regions have been separated, most systems then send the music regions and lyric regions to parallel OMR and OCR pipelines for fuller processing.

OCR itself is difficult for lyrics, however, because of inconsistent syllabification [6]. When sung over many notes, lyric words are usually (but not always) separated into their constituent syllables, which poses problems for

traditional OCR methods that rely on statistical language models with word dictionaries [7,8]. The need for syllabicated dictionaries limits the number of languages available for OCR and increases the number of unrecognised inflections [9]. Early documents add another twist on account of archaic spellings and scribal abbreviations [10]. Some systems sidestep these concerns by noting that the output of standard OCR on these documents is often good enough for users to be able to locate documents in a database [11], but for archival purposes, the ultimate goal of these systems is to have lyrics that are complete and accurate.

3. EDITING LYRICS WITH ARUSPIX

No automatic lyric recognition system will ever be perfect, however, and because digital archivists are among the primary target users for recognition software, these mistakes need to be corrected. These corrections can incur significant labour expenses in the absence of efficient software tools, to the point that, as we have seen in [11], sometimes they are not made at all. From the perspective of software design, then, an integrated editor for lyrics is a useful adjunct to any OMR package for vocal music. We chose to extend Aruspix, an open-source OMR application that already includes an integrated editor for musical symbols, to include a new editor for lyrics.

In the model underlying our lyric editor, lyrics are associated with notes in a many-to-one relationship, i.e., each musical note can be associated with one or more lyric elements. The user can modify these relationships and the lyrics themselves with a convenient graphical interface that pairs the editing region with the analogous portion of the original music image, as illustrated in Figure 1. Similar to the music editor in Aruspix, our lyric editor operates in one of two modes: Lyric Editing, which allows users to change the location of lyrics and their associated notes, and Lyric Insertion, which allows users to enter new lyrics and modify existing ones. Using this editor, any recognition errors can be identified and correctly quickly, which reduces labour costs for archival projects and facilitates rapid production of ground truth for researchers.

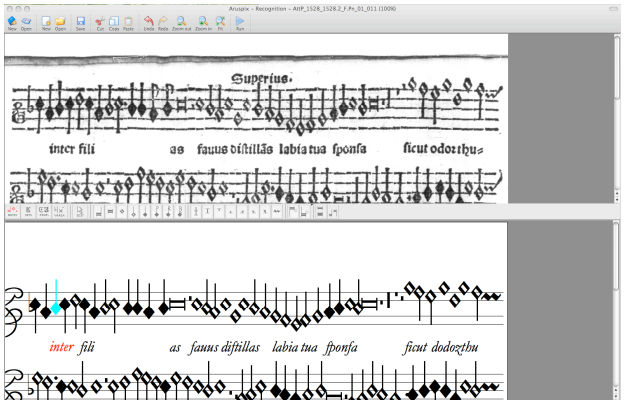


Figure 1. Screenshot of the lyric editor in Aruspix. The original image appears in the top pane. The lower pane contains the lyric editor. Each lyric is linked to a note; in this case, the word *inter* is linked to the third note on the top staff.

4. A METHOD FOR LYRIC-LINE EXTRACTION

Text-line detection is the first step in most OCR systems, and a great number of approaches have been developed for different types of documents: projection-based methods, grouping methods, and the Hough transform, among others [12]. Our lyric-line extraction algorithm, illustrated in Figure 2, is derived from an approach to text-line detection that is optimal for undulating lines [13,14]. Although lyrics are laid out less consistently than the text in text-only documents, they are almost always grouped along straight horizontal lines. After the removal of the staves, these straight lyric lines contrast sharply with the undulating lines, also detected by these algorithms, that trace the path of musical notes. More specifically, if baselines of both lyrics and notes are generated, the former will be almost straight while the latter will be highly curved and undulating. Thus, a line of lyrics can be extracted with confidence if many straight segments are found along the line. This assumption is fairly safe when there are a good number of words within a lyric line, e.g.,

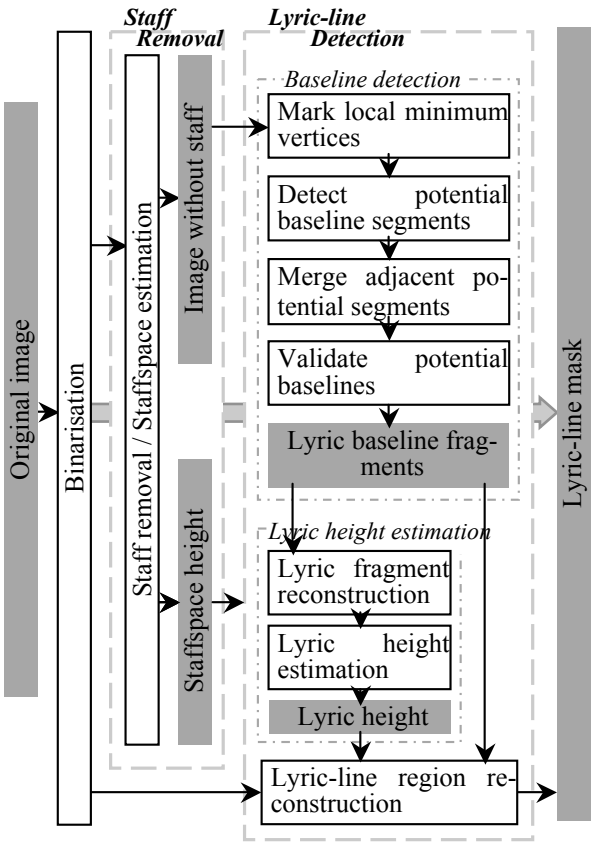
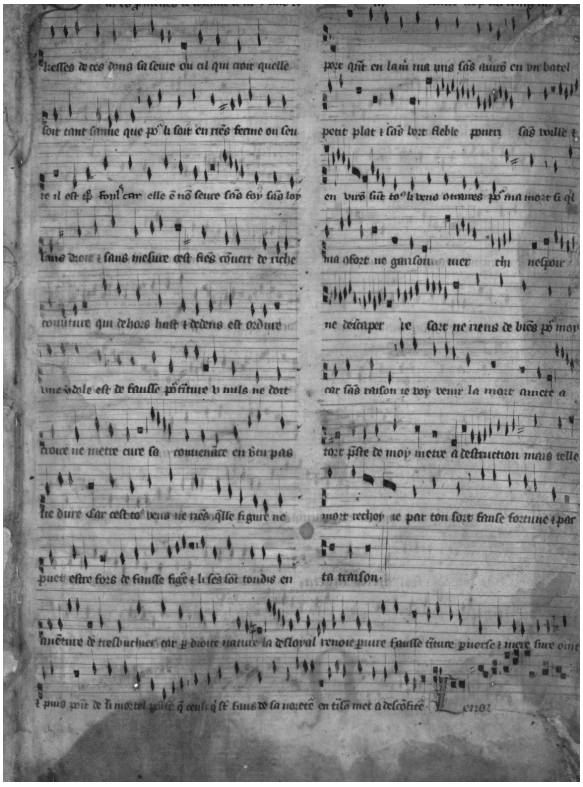
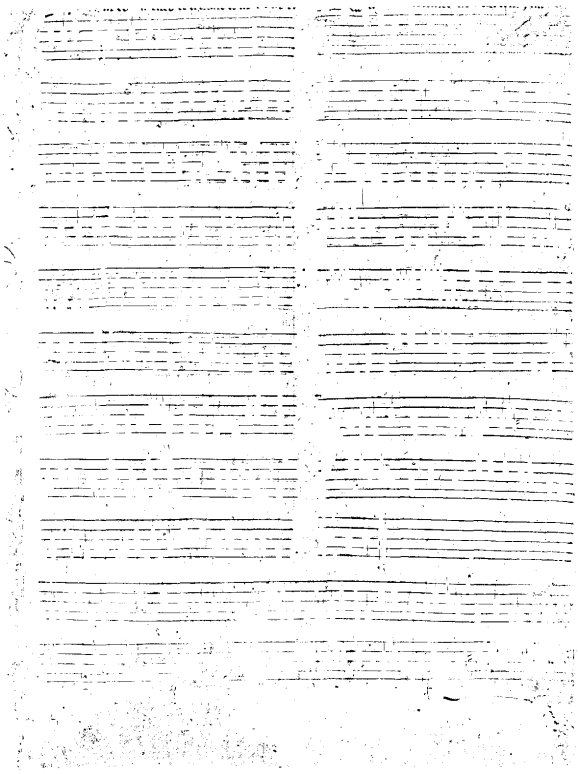


Figure 2. Workflow for extracting lyric lines. During preprocessing, the image is binarised and staves are removed. A three-step extraction process follows for lyrics.

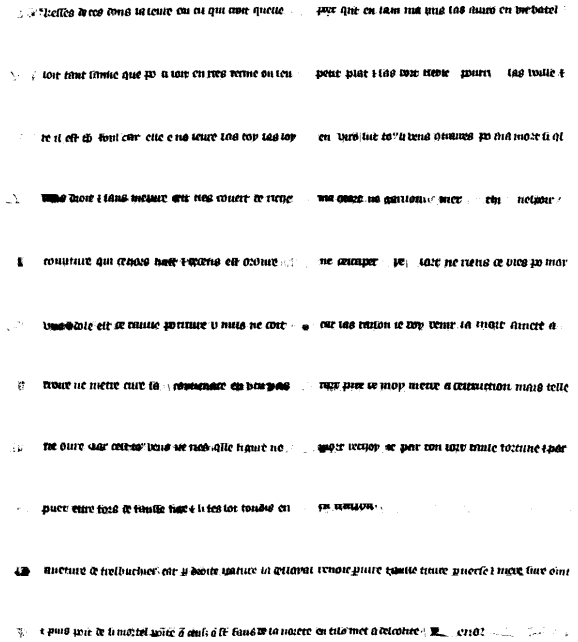


(a) Original image



(b) Reconstructed staff lines

(c) Local minima after staff-removal.



(d) Extracted lyric lines

Figure 3. Results of text-line extraction. Image (a), the original image, illustrates some of the layout challenges inherent to medieval documents. Reconstructed staff lines from our staff-removal step appear in (b). The "local minima" of each connected component appear in (c), and the final output of lyric-line extraction is in (d).

the lyric lines in Figure 3(a), but it can miss lyrics when they are arranged particularly sparsely.

Before detecting the lyric lines, the images must be preprocessed by global deskewing and staff removal. Because music recognition takes place in a distinct, parallel process, we developed a new staff-removal algorithm that damages noteheads slightly but removes staff lines more reliably in degraded documents than traditional techniques. The technique is in the style of the median-filter approach in [15]. Local horizontal projections and vertical run-length coding are used to estimate staff-space height and staff height. These values are used to construct a directional median-filter window in the form of a thin vertical bar. The bar is set to be tall enough to remove staff lines while remaining short enough to preserve notes and lyrics. Because the window is thin, this filter is able to remove curved staff lines effectively. Unlike some approaches, our algorithm is also able to locate the lyrics in documents without staff lines, e.g., early Aquitanian chant manuscripts; for such images, we obviously skip the staff-removal filter. A sample of staff lines extracted by this algorithm appears in Figure 3(b).

Following preprocessing, our method has three broad steps: baseline detection, estimation of lyric height, and reconstruction of lyric regions.

Baseline estimation begins by binarising the image (classifying pixels as foreground or background) and identifying all connected components of foreground pixels. Every component is represented by groups of vertices constituting the "local minima" of the component: the component is broken into vertical strips that are about as wide as a staff space is high, and the point closest to the bottom of the page is retained as a local minimum for the component. Figure 3(c) illustrates a chart of these local minima for one of our test images.

We then "connect the dots" to extract baselines. Each unconnected local minimum is connected to its nearest neighbour with respect to a quadratic thresholding function that weights the distance between the two points and the angle between them, privileging short distances and approximately horizontal lines (see Figure 4). This thresholding function is rather strict, and so unless lyrics are packed densely across a line, the connected baseline segments usually underscore individual words or letters rather than longer lyric lines. A second pass with a more permissive thresholding function is made to connect sufficiently long segments extracted from the first pass. Finally, all connected segments are validated to ensure that they contain a reasonable number of local minima and have an overall horizontal slope. This final validation step removes any segments that might arise from musical lines with repeated notes or sequences of notes that are close in pitch space.

In extracting the baselines, we discard all information about the height of the lyrics, and so it is necessary to reconstruct lyric regions from the baselines. The process is the inverse of marking the local minima: for each local minimum identified in a baseline, the corresponding connected component is included in the lyric region. Some of these connected components, however, include non-lyric elements, especially when lyric elements close to a staff overlap with low notes. In order to compensate for these problems, an upper bound for lyric height is chosen based on the size of a staff space. Connected components above this upper bound are cropped.

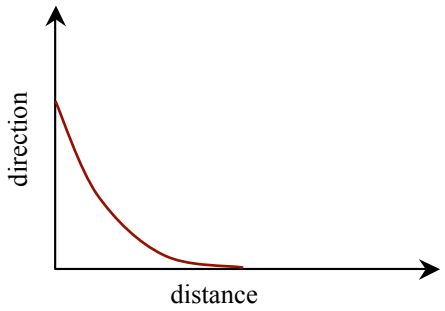


Figure 4. Thresholding function for connecting local minima of the connected components. The boundary is quadratic and privileges horizontal directions.

After the lyric height has been estimated, complete lyric-line regions are generated from the baselines. In the absence of other information about page layout, the lyric lines are extended from the left-most to the right-most points of the page. A simple peak-picking process along the y-axis groups extracted baselines that are part of common lyric lines; the peak picking is tuned with the estimated lyric height as described above. In our experiments, a simple linear regression on the baselines combined with estimated lyric height yields good lyric regions in most cases, although when the pages are non-linearly skewed, higher-order polynomials are necessary.

We tested our algorithm on a set of 40 images from the Digital Image Archive of Medieval Music (DIAMM) chosen for their particularly challenging layouts [16]. A sample image is presented in Figure 3(a), and the output of our algorithm on this image is presented in Figure 3(d). Note in particular that the algorithm proved robust to the two-column format. Despite the challenging layouts and (in some cases) considerable document degradation, our algorithm was able to recall 80.4 percent of the text lines with 88.4 percent precision overall. For clean images, however, the results are nearly perfect: there was only one recall error across our 12 cleanest samples with 100-percent precision.

5. SUMMARY AND FUTURE WORK

Automatic lyric recognition is challenging for any musical document on account of varied page layouts and inconsistent syllabification. This challenge is exacerbated for early music documents, which suffer from an even wider variety of layouts and, often, significant degradation. We have developed software that helps researchers generate ground truth quickly for early music documents

and that helps archivists correct any errors in automatic recognition with minimum labour cost and musical sense. We have also extended approaches for document image analysis for historical text documents to be sensitive to the particularities of music manuscripts, resulting in reliable text-line extraction from a number of difficult older documents.

This work is in progress, and we are currently using data we have extracted from these systems to experiment with full-scale recognition using features similar to those in [7] and a variety of labelling models, including hidden Markov models [17] and conditional random fields [18]. The long-term goal of our project is to produce a set of extensions to Aruspix that will make it a fully functional OMR system for early vocal music, including OMR for medieval plainchant notations and automatic lyric recognition for the most common languages of the period.

6. ACKNOWLEDGEMENTS

The authors would like to thank the Canada Foundation for Innovation (CFI) and the "Image, Text, Sound, and Technology" program of the Social Sciences and Humanities Research Council of Canada (SSHRC) for supporting this research.

7. REFERENCES

[1] Pinto, J. C., P. Vieira, M. Ramalho, M. Mengucci, P. Pina, and F. Muge. 2000. Ancient music recovery for digital libraries. In *Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries*, 24–34.

[2] Ouyang, Y., J. A. Burgoyne, L. Pugin, and I. Fujinaga. 2009. A robust border-detection algorithm with application to medieval manuscripts. In *Proceedings of the International Computer Music Conference*.

[3] Pugin, L. 2006. Optical music recognition of early typographic prints using hidden Markov models. In *Proceedings of the 7th International Conference on Music Information Retrieval*, 53–6.

[4] Choudhury, G. S., T. DiLauro, M. Droettboom, I. Fujinaga, and K. MacMillan. 2001. Strike up the score: Deriving searchable and playable digital formats from sheet music. *D-Lib Magazine* 7 (2).

[5] Jones, G., B. Ong, I. Bruno, and K. Ng. 2007. Optical music imaging: Music document digitization, recognition, evaluation. In *Interactive Multimedia Music Technologies*, ed. K. Ng and P. Nesi, 50–79. Hershey, PA: IGI Global.

[6] Droettboom, M. 2004. Beyond transcription: Case studies in special document analysis requirements. In *Proceedings of the International Workshop on Document Image Analysis for Libraries*.

[7] Vinciarelli, A., S. Bengio, and H. Bunke. 2004. Offline recognition of unconstrained handwritten texts using HMMs and statistical language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (6): 709–20.

[8] Steinherz, T., E. Rivlin, and N. Intrator. 1999. Offline cursive script word recognition: A survey. *International Journal on Document Analysis and Recognition* 2 (2–3): 90–100.

[9] Wingenroth, B., M. Patton, and T. DiLauro. 2002. Enhancing access to the Levy sheet music collection: Reconstructing full-text lyrics from syllables. In *Proceedings of the ACM-IEEE Joint Conference on Digital Libraries*, 308–9.

[10] Ernst-Gerlach, A., and N. Fuhr. 2007. Retrieval in text collections with historic spelling using linguistic and spelling variants. In *Proceedings of the ACM-IEEE Joint Conference on Digital Libraries*, 333–41.

[11] Diet, J., and F. Kurth. 2007. The Probado music repository at the Bavarian State Library. In *Proceedings of the 8th International Conference on Music Information Retrieval*, 501–4.

[12] Likforman-Sulem, L., Z. Abderrazak, and T. Bruno. 2007. Text line segmentation of historical documents: A survey. *International Journal on Document Analysis and Recognition* 9 (2): 123–38.

[13] Feldbach, M. and K. D. Tonnies. 2001. Line detection and segmentation in historical church registers. In *Proceedings of the 6th International Conference on Document Analysis and Recognition*, 743–7.

[14] Pu, Y., and Z. Shi, Z. 1998. A natural learning algorithm based on Hough transform for text lines extraction in handwritten documents. In *Proceedings of the 6th International Workshop on Frontiers in Handwriting Recognition*, 637–46.

[15] Fornes, A., J. Llados, and G. Sanchez. 2005. Staff and graphical primitive segmentation in old handwritten scores. In *Artificial Intelligence Research and Development*, ed. B. López, J. Meléndez, P. Radeva, and J. Vitrià, 83–90. Amsterdam: IOS Press.

[16] Digital Image Archive of Medieval Music. http://www.diamm.ac.uk/ (accessed 22 May 2009).

[17] Artières, T., N. Gauthier, P. Gallinari, and B. Dorizzi. 2002. A hidden Markov models combination framework for handwriting recognition. *International Journal on Document Analysis and Recognition* 5 (4): 233–43.

[18] Lafferty, J., A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*, 282–9.