STAFF LINE DETECTION AND REMOVAL WITH STABLE PATHS

Artur Capela, Ana Rebelo
INESC Porto, Campus da FEUP, Rua Dr. Roberto Frias 378, 4200-465 Porto, Portugal
gcapela@inescporto.pt, arebelo@inescporto.pt

Jaime S. Cardoso
INESC Porto, Faculdade de Engenharia, Universidade do Porto, Portugal
jaime.cardoso@inescporto.pt

Carlos Guedes
INESC Porto, Escola Superior de Música e Artes do Espectáculo, Portugal
carlosguedes@mac.com

Keywords: Music, optical character recognition, document image processing, image analysis.

Abstract: Many music works produced in the past are currently available only as original manuscripts or as photocopies. Preserving them entails their digitalization and consequent accessibility in a machine-readable format, which encourages browsing, retrieval, search and analysis while providing a generalized access to the digital material. Carrying this task manually is very time consuming and error prone. While optical music recognition (OMR) systems usually perform well on printed scores, the processing of handwritten music by computers remains below the expectations. One of the fundamental stages to carry out this task is the detection and subsequent removal of staff lines. In this paper we integrate a general-purpose, knowledge-free method for the automatic detection of staff lines based on stable paths, into a recently developed staff line removal toolkit. Lines affected by curvature, discontinuities, and inclination are robustly detected. We have also developed a staff removal algorithm adapting an existing line removal approach to use the stable path algorithm at the detection stage. Experimental results show that the proposed technique outperforms well-established algorithms. The developed algorithm will now be integrated in a web based system providing seamless access to browsing, retrieval, search and analysis of submitted scores.

1 INTRODUCTION

The Universal Declaration on Cultural Diversity adopted by the General Conference of UNESCO on 2001 asserts that cultural diversity is as necessary for humankind as biodiversity is for nature, and that policies to promote and protect cultural diversity thus are an integral part of sustainable development. Being music a pivotal part of our cultural heritage, its preservation, in all of its forms, must be pursued. Frequently, the preservation of many music works entails their digitalization and consequent accessibility in a format that encourages browsing, analysis and retrieval.

There is a vast amount of invaluable paper-based heritage, including printed and handwritten music scores, which are deteriorating over time due to natural decaying of paper and chemical reaction (i.e., between written ink and paper). Various efforts have been focused on this issue in order to preserve the record of our heritage. Digitisation has been commonly used as a possible tool for preservation. Although a digital copy may not conserve the original document, it can preserve the most important part: its data. It has also the advantages of easy duplications, distribution, and digital processing. Nevertheless, the output of the digitalization process is not amenable for further analysis or semantic search operations. Thus, an Optical Music Recognition (OMR) process is needed. However, the manual process required to recognize handwritten musical symbols in scores and to put them in relationship with the spine structure of the score is very time consuming. This justifies the research around reliable automatic OMR algorithms as current solutions are still below the expectations.

As a concrete example, Portugal has a notorious lack in music publishing from virtually all eras of its musical history. However, whereas most of the known original music manuscripts before the twentieth century are kept at the National Library Archive in Lisbon, there is virtually no national repository for the Portuguese music from the twentieth century. Although there are recent efforts in order to catalogue and preserve in digital form the Portuguese music from the late twentieth century—notably the Music Information Center (MIC, 2008) and the section on musical heritage from the Institute of the Arts website (IOA, 2008)—most of the music pre-dating computer notation software was never published and still exists as manuscripts or photocopies spread out all over the country in inconspicuous places. The risk of irreversibly losing this rich cultural heritage is thus a reality.

1.1 OMR System

The project "Optical recognition system for handwritten music scores" initiated in 2007 by INESC Porto and ESMAE is the point of departure for creating a web-based system of music manuscripts of Portuguese composers from the twentieth century. This database will provide generalized access to a wide *corpus* of unpublished handwritten music encoded in MusicXML, which can be accessed remotely via the Internet. The database will not only centralize as much information as possible but will also serve to preserve this *corpus* in a way that is easily accessible for browsing, analysis, and ultimately, for performing this *repertoire*, therefore helping to keep the Portuguese music alive (Capela et al., 2008). Although the aim of this project is Portuguese music, it is equally valid for all printed and handwritten music scores that need to be preserved from all around the world.

The ambitious goal of providing generalized access to handwritten scores that have never been published has been severely hampered by the actual state-of-the-art of handwritten music recognition. There are currently various commercial OMR software solutions (Capella software[1], SharpEye Music Reader[2], OMeR[3]) and a few open source solutions (AOMR2[4], OpenOMR[5], Audiveris[6]), but they are all offline standalone applications. The existing online archives of music scores (Lester Levi Collection, 2008; Classical Sheet Music Collection, 2008; Mutopia Collection, 2008) usually provide them in inadequate formats—usually only as the scanned score image—for retrieval or automatic analysis. These online archives are mere standard websites, without facilities for optical recognition, editing and searching through the scores musical content. The creation of an OMR system, integrating optical recognition, storage, search, browsing and downloading capabilities, while keeping the scores in their original format along with their digital counterpart, would therefore be extremely beneficial. An integrated score editor would be provided in order to view and edit the submitted music scores.

In our previous work on this project we have presented a complete OMR System solution—OMRSYS (Capela et al., 2008)—comprising a database driven web application with one or more OMR applications integrated in the proposed system. The proposed architecture successfully attends the stated objectives. At the end of our project we plan on developing a fully functional system according to the specified architecture and integrating a complete OMR package.

1.2 Detection and Removal of Staff Lines

Staff line detection and removal are the first fundamental stages on the OMR process, with subsequent processes relying heavily on their performance. The reasons for detecting and removing the staff lines lie on the need to isolate the musical symbols for a more efficient and correct detection of each symbol present on the score. Although their primary application is as a preprocessing step in the recognition of music notation, the line detection problem also occurs in different contexts (e.g., the recognition of bank transfer forms).

The detection of staves is complicated due to a variety of reasons. The handwritten staff lines are rarely straight and horizontal, and are not parallel to each other. For example, some staves may be tilted one way or another on the same page or they may be curved. These scores tend to be rather irregular and determined by a person's own writing style. Moreover, if we consider that most of these works are old, the quality of the paper in which it is written might have degraded throughout the years, making it a lot harder to correctly identify its contents.

In (Cardoso et al., 2008a; Cardoso et al., 2008b) we presented a new and robust staff line detection algorithm based on a stable paths approach. The proposed paradigm uses the image as a graph, where the staff lines result as connected paths between the two lateral margins of the image. A staff line can be considered as a connected path from the left side to the

[1]http://www.capella-software.com/capscan.htm

[2]http://www.visiv.co.uk

[3]http://www.myriad-online.com/en/products/omer.htm

[4]http://www.bzzt.net/~arnouten/wiki/index.php/Gamera#AOMR2:_omr_toolkit

[5]http://sourceforge.net/projects/openomr

[6]http://audiveris.dev.java.net

right side of the music score. As staff lines are almost the only extensive black objects on the music score, the path to look for is the shortest path between the two margins if paths (almost) entirely through black pixels are favoured.

In this paper we present our recent work and results focusing on the implementation of the Stable Paths algorithm as a C++ plugin for the MusicStaves Toolkit (Dalitz et al., 2008; MusicStaves Toolkit, 2008) (based on the Gamera Framework (MacMillan et al., 2002; Gamera Framework, 2008)), as well as on the removal stage by using our detection algorithm in the first stage. We have also adapted a removal algorithm based on the LineTrack Height approach proposed in (Dalitz et al., 2008), which we will present in section 3. In section 2 the Gamera and MusicStaves Toolkit are presented, together with our C++ plugin. In section 4, both our proposed detection and removal algorithms are evaluated experimentally using a well-known dataset of music scores. Finally, conclusions are drawn and future work is outlined in section 5.

2 STABLE PATHS INTEGRATION

In this section we present the platform in which our detection algorithm was integrated for testing and validation. The platform is comprised by its core—the Gamera Framework (MacMillan et al., 2002; Gamera Framework, 2008)—and by a toolkit for the evaluation of detection and removal algorithms—the MusicStaves Toolkit (Dalitz et al., 2008; MusicStaves Toolkit, 2008). This toolkit is a set of Gamera plugins aiming to support the development and test of staff line detection and removal algorithms for music scores, by extending the Gamera functionality.

After we describe each part constituting this platform, we present the implementation of our staff line detection algorithm as a C++ plugin on the MusicStaves Toolkit. Finally, we present the integration of our detection algorithm on some removal algorithms in the MusicStaves toolkit. We have integrated our algorithm in those removal algorithms where the detection is processed separately from the removal operation, replacing the Dalitz algorithm (Dalitz et al., 2008) as the detection stage.

2.1 Gamera Framework

Gamera (MacMillan et al., 2002; Gamera Framework, 2008) is a portable and open source framework to create structured documents analysis applications by domain experts. The name Gamera is an acronym to "Generalized Algorithms and Methods for Enhancement and Restoration of Archives". It combines a programming library with graphical tools for an interactive training and development of recognition systems. This framework tries to be a tool to create custom applications through the domain experts knowledge instead of responding to various requirements with a monolithic application. It aims at providing an efficient test and refinement development cycle. The programming language in its basis is the high-level language Python, although it has many extensions written in C++ to carry out low-level image processing. Nevertheless, due to its nature, Python turns the code writing process more agile and facilitates the use of scripting, which makes Gamera an interactive and batchable framework. Besides the large number of extensions deployed with this framework, it is also possible to customize and extend with plugins or toolkits, written in Python and C++.

The Gamera framework is modular and organized in a series of horizontal layers, which can be seen at Figure 1.

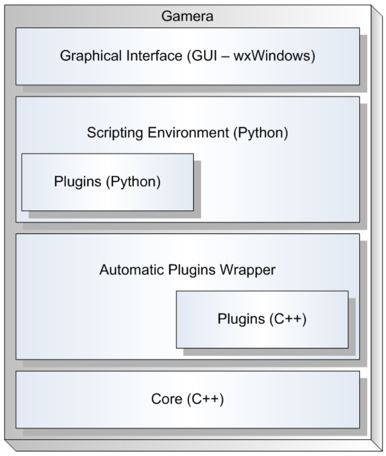


Figure 1: Gamera Architecture.

Gamera follows a modular plugin architecture. It is made of modules (plugins), both written in Python and C++, integrated in a high-level scripting environment. Each module executes a task on the recognition process. The framework maintains a toolbox design approach, i.e., a user has access to a large set of tools for the optical recognition stages.

2.2 MusicStaves Toolkit

MusicStaves (Dalitz et al., 2008; MusicStaves Toolkit, 2008) is a Gamera toolkit specific for the development and test of staff line detection and removal algorithms. This Phyton toolkit also integrates facilities to create a test set of music scores and to evaluate results with established metrics. As with Gamera, this toolkit is portable and its source code is freely available. In order to use MusicStaves, it must be imported to Gamera, either in the GUI or in a programmatic manner.

MusicStaves is structured as seen in Figure 2. The toolkit is composed by a set of main classes where the staff line detection and removal algorithms are implemented, and by a set of plugins in Python and C++. Some plugins are used by the implemented algorithms while others are tools to aid the algorithms testing. It is an extendable toolkit in which one can integrate a new staff line detection and/or removal algorithm. Its plugin system follows the Gamera framework rationale and as such the plugins may be written in Python and C++. In order to write new staff line detection or removal algorithms the toolkit provides two interfaces: *StaffFinder* (detection) and *MusicStaves* (removal).

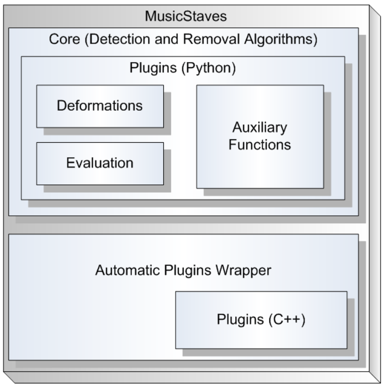


Figure 2: MusicStaves Architecture.

A test set of 32 synthetic music scores is also provided by the authors of this toolkit. Moreover, the toolkit allows applying a set of deformations (e.g., rotation, curvature, typeset emulation, white speckles) commonly found in the real world to these perfect scores—see Figure 3. The purpose is to be able to measure the performance of the removal algorithms contained in MusicStaves by using three defined error metrics (Dalitz et al., 2008): Pixel Level, Segmentation Region Level and Staffline Interruptions. However, the same set may be used to evaluate staff line detection algorithms alone by defining adequate error metrics (Cardoso et al., 2008a; Cardoso et al., 2008b). We have restricted the range of the parameters controlling the intensity of the deformations to values considered realistic.

2.3 Integration

We have integrated our recently proposed algorithm for staff line detection (Cardoso et al., 2008a; Cardoso et al., 2008b) in MusicStaves as a C++ plugin.

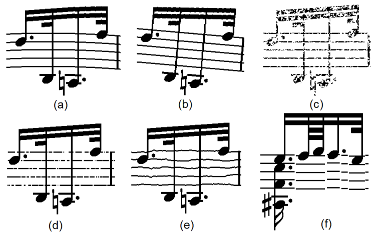


Figure 3: Some examples of applied deformations: a) Curvature; b) Rotation; c) White Speckles; d) Staffline Interruptions; e) Staffline y-variation; f) Typeset Emulation. See (Dalitz et al., 2008) for details.

The plugin encompasses the main *StaffFinder* class in the toolkit root, the respective Python plugin in the Python plugins folder and the algorithm implementation in C++ in the C++ plugins folder. In Figure 4 we present a diagram of the algorithm as integrated in MusicStaves.

The algorithm processing starts with the call to the method *find_staves*, which receives a binarized image as input. That image is then passed to the function in C++ by the plugin Python code. In the C++ implementation, after the respective staff line detection function is called, the image is converted to the format used internally by our algorithm. After the whole detection task is complete it returns the staff lines skeleton in the MusicStaves format. After receiving the skeleton list on its Python class, it fills the structure *self.linelist* with the obtained values.

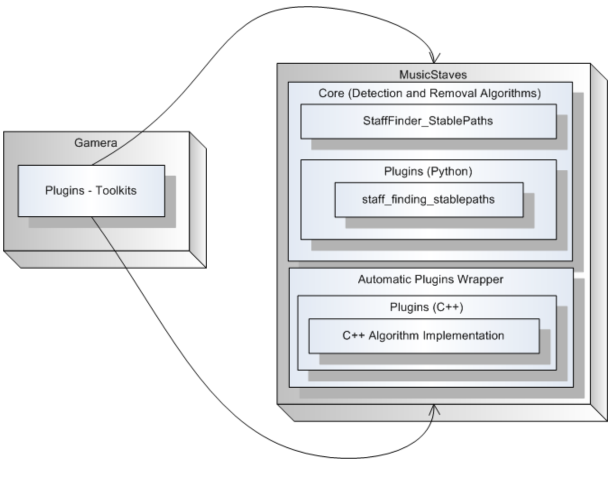


Figure 4: Overall view.

Besides integrating our Stable Paths Approach algorithm into the MusicStaves toolkit as a C++ plugin, we have also integrated the algorithm with staff removal algorithms in order to evaluate the improvements over the original results. However, this is not a standard integration as the toolkit does not provide the means for this kind of integration. This integration was coded in the staff removal algorithms by adding the possibility to choose the staff line detection algorithm through a parameter value. A diagram illustrating this integration can be seen in Figure 5. Some removal algorithms present in MusicStaves detect the staff lines along with the removal process. From those, we have used the one with best results in general (Dalitz et al., 2008)—Skeleton—for comparison purposes.

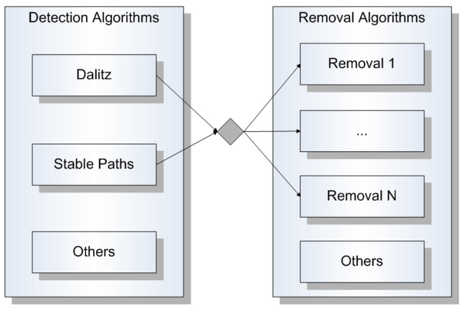


Figure 5: Stable Paths integration with staff removal algorithms.

3 STAFF REMOVAL ALGORITHMS

In the process of recognizing music scores the staff removal algorithm is processed after the staff line detection stage takes place. In our current work the removal algorithm is based on the LineTrack Height algorithm presented on (Randriamahefa et al., 1993). The goal on this method is to track the staff lines positions obtained by a detection algorithm and remove vertical run sequences of black pixels that have a value lower than a specified threshold, which was experimentally set at *2*staffHeight*.

As music scores may suffer from deformations, the staff lines may have descontinuities, be curved or inclined. These problems will influence the success to achieve a correct detection of lines contained on the score to recognize. However, due to the abovementioned problems, the positions of the staff lines obtained by a staff line detection algorithm may pass slightly above or under the real staff lines positions. That way, if we are in presence of a white pixel when the staff lines are tracked, we search vertically for the closest black pixel. If that distance is lower than a specified tolerance—experimentally chosen as *1+ceil(staffHeight/3.0)*—we move the reference position of the staff line to the position of the black pixel found.

On (Dalitz et al., 2008) a new method is presented—Skeleton—which uses the skeleton information, but performs the staff removal on the original image instead of the skeleton. The method relies on the fact that symbols on the stafflines lead to junction points or corner points in the skeleton.

Algorithm 1 Staff Removal Algorithm.

procedure STAFFLINEREMOVAL($IMAGE$,$STAVES$)
$threshold = 2*$staffHeight;
  $tolerance = 1 + ceil($staffHeight$/3.0)$;
  $IMAGE\_REMOVE = copy(IMAGE)$;
  for $nvalid = 0$ to STAVES size **do**
    $Point2D$ staff $= validStaves[nvalid]$;
    for $i = 0$ to staff size **do**
      $col =$staff$[i].x$;
      $refRow =$staff$[i].y$;
      $row = refRow$;
      $pel = valuePixel(IMAGE, IMAGE\_REMOVE)$
      decrement/increase the reference row until one pixel different from white pixel (dist1/dist2) is found;
      if $dist1 \leq max(1, min(dist2, tolerance))$ **then**
        $refRow- = dist1$;
      else
        if $dist2 \leq max(1, min(dist1, tolerance))$ **then**
          $refRow+ = dist2$;
        else
          continue;
        end if
      end if
      Count the number of decrements/increase on the reference row until the black pixel changes to white pixel (run);
      if $run \geq threshold$ **then**
        continue;
      end if
      remove the vertical black sequences on the IMAGE;
    end for
  end for
end procedure

4 RESULTS

Although the evaluation of new staff detection algorithms may be done by visually inspecting the output on a set of scores—as adopted on (Rebelo et al., 2007)—, our current comparison is supported on quantitative measures. The test set adopted for the qualitative evaluation of the proposed method is the one presented in (Dalitz et al., 2008) and already described. It consists of ideal synthetic scores to which a set of known deformations have been applied—see (Dalitz et al., 2008) for more details. In total we have generated 2688 deformed images originated from 32 perfect scores. In order to conveniently measure the performance of staff line removal algorithms we have adopted two error metrics from (Dalitz et al., 2008): Pixel Level and Segmentation Region Level.

Staff line detection algorithms can be used as a first step in many staff removal algorithms. To understand the potential of our algorithm to leverage the performance of existing staff removal algorithms, we conducted a series of experiments, comparing the original version of a staff removal algorithm with the modified version of it, making use of the Stable Paths algorithm at the staff line detection step. The quantitative comparison of the different algorithms is totally in line with the comparison presented in (Dalitz et al., 2008).

With respect to the considered distortions, regarding the detection stage, the Stable Paths based approach outperforms the Dalitz algorithm. In Figure 6 we present our results for the removal algorithms: LineTrack Height (with Dalitz and Stable Paths), Skeleton and LineTrack Height Modified (with Stable Paths). We chose the methods that present the best results in (Dalitz et al., 2008), implementing our own removal algorithm with LineTrack Height as a basis. In general we verify that the replacement of the Dalitz method by our Stable Paths Approach algorithm as the staff detection step has improved the final staff line removal results[7]. Additionally, the LineTracking Height Modified algorithm presents an overall better performance than the original LineTrack Height algorithm from (Dalitz et al., 2008). Our staff line detection and removal approaches also outperform the Skeleton method, although it continues to present a competitive performance. We have not integrated the Stable Paths algorithm with the Skeleton algorithm as the second performs the lines detection along with their removal instead of using two separate stages. All the parameters, both on the Stable Paths detection algorithm and LineTrack Height Modified, were preliminary tuned over an independent set of images.

This performance gain is even more noteworthy as the MusicStaves algorithms are receiving as input the correct number of lines per staff. Had not this been the case, the differential between both would have been much larger. In summary, these experiments show the strength of the algorithms presented here. Despite being based on simple and intuitive underlying principles, the performance of the proposed algorithms is quite competitive.

The analysed results have covered the detection and removal accuracy but a brief word on speed is also in order. Comparing different algorithms for speed is notoriously difficult; we are simultaneously judging mathematical properties and specific implementations. In the experimental study, the current implementation of the Stable Paths algorithm run almost as fast as the Dalitz algorithm (20% slower). In respect to the removal algorithms our LineTrack Height version with Stable Paths is significantly faster than the Skeleton algorithm (two times faster). Comparing to the original LineTrack Height algorithm with the Dalitz detection algorithm the runtime difference is not significant. The algorithms were evaluated as available at the Staff Removal Toolkit (Dalitz et al., 2008).

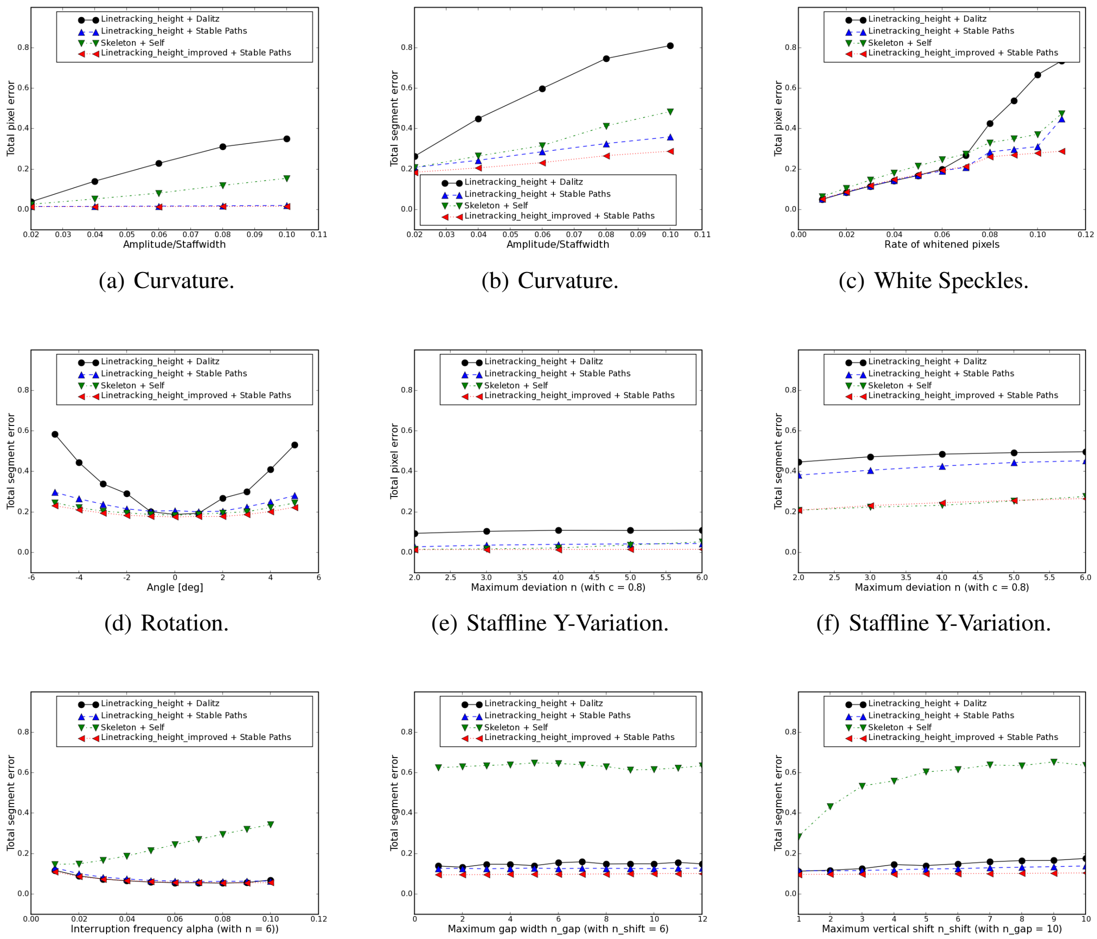
[7]For the deformations not shown, the stable path is not significantly better than Dalitz.

5 CONCLUSIONS

This paper presented the integration of our robust Stable Paths Approach algorithm (Cardoso et al., 2008a; Cardoso et al., 2008b) in the MusicStaves Toolkit (Dalitz et al., 2008) as a C++ plugin, an improved version of an existing staff line removal algorithm—LineTrack Height (Dalitz et al., 2008), and the results we have obtained in our staff line removal tests. We have integrated our detection algorithm with existing staff line removal algorithms. Our approach successfully deals with the difficulties posed by the symbols superimposed on the staff lines as well as a wide range of image conditions (e.g., discontinuities, curved lines), frequently found on handwritten scores.

The encouraging results lead us now to consider investigating the detection of music symbols benefiting from the improved staff line detection and removal, creating a complete OMR application in order to integrate it on our proposed complete OMR solution—OMRSYS (Capela et al., 2008). Thus, our proposed system offers a complete solution for the preservation of our musical heritage. It includes an optical recognition engine integrated with an archiving system and a user-friendly interface for searching, browsing and edition. The digitized scores are stored in MusicXML, a recent and expanding music interchange format designed for notation, analysis, retrieval, and performance applications.

Our proposed algorithms and complete OMR system promote the creation of a full *corpus* of music documents, promoting its preservation and study. This project will culminate in the creation of a repository of handwritten scores, accessible online. The database will be available for enjoyment, educational and musicological purposes, thus preserving this *corpus* of music in an unprecedented way.



(a) Curvature. (b) Curvature. (c) White Speckles.

(d) Rotation. (e) Staffline Y-Variation. (f) Staffline Y-Variation.

(g) Staffline Interruptions. (h) Typeset Emulation Part 1. (i) Typeset Emulation Part 2.

Figure 6: Effect of different deformations on the overall staff removal error rates. See (Dalitz et al., 2008) for parameter details.

ACKNOWLEDGEMENTS

This work was partially funded by Fundação para a Ciência e a Tecnologia (FCT) - Portugal through project PTDC/EIA/71225/2006.

REFERENCES

Capela, A., Cardoso, J. S., Rebelo, A., and Guedes, C. (2008). Integrated recognition system for music scores. In *International Computer Music Conference (ICMC) (Accepted)*.

Cardoso, J. S., Capela, A., Rebelo, A., and Guedes, C. (2008a). A connected path approach for staff detection on a music score. In *International Conference on Image Processing (ICIP 2008) (Accepted)*.

Cardoso, J. S., Capela, A., Rebelo, A., and Guedes, C. (2008b). Staff detection with stable paths. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (Submitted)*.

Classical Sheet Music Collection (2008). Classical Sheet Music and MIDI Files, `http://www.music-scores.com`.

Dalitz, C., Droettboom, M., Pranzas, B., and Fujinaga, I. (2008). A comparative study of staff removal algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):753–766.

Gamera Framework (2008). Gamera Framework, `http://ldp.library.jhu.edu/projects/gamera/`.

IOA (2008). Institute of the Arts, `http://patrimonio.dgartes.pt/?lang=pt`.

Lester Levi Collection (2008). The Lester S. Levi Collection of Sheet Music, `http://levysheetmusic.mse.jhu.edu/`.

MacMillan, K., Droettboom, M., and Fujinaga, I. (2002). Gamera: Optical music recognition in a new shell. In *International Computer Music Conference (ICMC)*, pages 482–485.

MIC (2008). Music Information Center, `http://www.mic.pt`.

MusicStaves Toolkit (2008). MusicStaves Toolkit, `http://lionel.kr.hsnr.de/~dalitz/data/projekte/stafflines/`.

Mutopia Collection (2008). The Mutopia Project, `http://sca.uwaterloo.ca/Mutopia`.

Randriamahefa, R., Cocquerez, J. P., Fluhr, C., Ppin, F., and Philipp, S. (1993). Printed music recognition. In *Proceedings of the 2nd International Conference on Document Analysis and Recognition (ICDAR'93)*, pages 898–901.

Rebelo, A., Capela, A., da Costa, J. F. P., Guedes, C., Carrapatoso, E., and Cardoso, J. S. (2007). A shortest path approach for staff line detection. In *AXMEDIS '07: Proceedings of the Third International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution*, pages 79–85, Washington, DC, USA. IEEE Computer Society.