FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Staffline detection and removal in the Grayscale Domain

Vítor C. Vidal

Mestrado Integrado em Engenharia Electrotécnica e de Computadores

Supervisor: Jaime S. Cardoso (Ph.D.)

Co-Supervisor: Ana Maria Rebelo (MSc)

26 June 2012

Abstract

In order to preserve the great musical pieces of the past which were originally handwritten, several databases containing digitized and photocopied versions of these documents were created, simplifying the storage and sharing of these documents. Over the years, the need to create new methods for indexing these files started to emerge, leading to the creation of the Optical Musical Recognition (OMR) engines. The main objective of these engines is to read the music scores information using digitized documents. This reading implies the understanding of the music score content in a way that allows the navigation in these databases, not only by composers, titles, date, size, etc. but also by melody. These engines will also be capable of reproducing the manuscripts information in a digital audio format.

The detection of stafflines is the first step of most OMR processes. Its great significance derives from the ease with which we can then proceed with the extraction of musical symbols. This step is usually achieved using the binarization of images by setting thresholds that can be local or global. These techniques however, can remove some relevant information of the music sheet and introduce noise in the later stages of the process. It arises therefore a need to create a method that eliminates the loss of information due to the binarization. Throughout this dissertation, the "Staff Detection with Stable Paths" [1] method, that works based on binary images, will be adapted to work with grayscale images. In order to do that, a new weight function that applies a value to each pair of pixels based on the average of their luminance values will be implemented. To improve the results, a function that estimates the range of luminance values of the stafflines will also be created.

Acknowledgements

The dissertation is undoubtedly the high point of a master degree student. It is the moment to show what we have learned, the moment to develop some significant work and most of all, a moment to take a taste of what we can do, of what are our limitations and how can we surpass them. A moment to understand that we can do whatever we dare to dream, if we have people willing to help us. Since I was fortunate with people like this I would like to acknowledge them.

First of all I am grateful to all my teachers that, since I were a little kid, taught me to embrace the knowledge. I must thank INESC TEC (formerly INESC Porto) for the place and resources it provided me, and more specifically to the visual computing and machine intelligence (VCMI) group, that welcomed me and broadened my horizons.

There are also two people who were essential for the development of my dissertation that I must mention. Professor Jaime Cardoso, for his guidance and all the great ideas and concepts that significantly improved my work performance and Ana Rebelo, for all the time she spent helping me in my work, not only correcting, teaching and guiding me, but also motivating and creating a closer bond that eased our interactions.

I also have to thank to all my family, specially to my parents, for the support they gave me all this time, for all the patience and good counsels and for what they gave up so I could get to this point. To Nini, for being an example to me and to Mónica, for all the encouraging words and for standing by my side at the most difficult times.

At last, I have to mention all my friends for easing my concerns.

Vítor C. Vidal

"Choose a job you love,
and you will never have to work a day in your life."

Confucius

Contents

List of Figures

List of Tables

Abbreviations

OMR	Optical Music Recognition
BLIST	Binarization based in LIne Spacing and Thickness
RLE	Run-length encode
DAG	Directed Acyclic Graph
dpi	Dots per inch
MIDI	Musical Instrument Digital Interface
MusicXML	Music Extensible Markup Language
StudECE	Students Conference in Electrical and Computer Engineering
IMSLP	International Music Store Library Project
ICDAR	International Conference on Document Analysis and Recognition

Chapter 1

Introduction

One of the problems that we face today is the deterioration and loss of information over time of unique printed and handwritten manuscripts. Musical scores are among such perishable documents. The digitization has been used to preserve these data, but to convert the paper-based music scores into a machine-readable format to facilitate operations such as search, retrieval and analysis, an Optical Music Recognition (OMR) system is needed. In these algorithms, the detection and removal of the stafflines are fundamental to isolate the musical symbols and to have a more efficient and correct detection of each symbol. Currently, all the OMR methods need to convert the image from grayscale to binary, which results in a possible loss of indispensable information. This work introduces an alternative to all these methods by working over the original grayscale image, which result on more accurate results even on damaged images.

1.1 Context

Nowadays there are some libraries, such as the International Music Store Library Project (IMSLP)[1] that possess documents that are already indexed to the databases not only through parameters of usual researches, but also throughout its own melody. The existence of documents in these formats, its mainly due to the work of people that transcript the documents manually, or recently, due to developed image processing techniques that compile the information in an automatic way (cf. chapter 3.1 about Optical Music Recognition – OMR).

The method to be developed thorough this essay is part of the "Automatic recognition of handwritten music scores" project, which aims to create an automatic OMR that facilitates the creation of future repositories of musical pieces. Methods that allow the detection of handwritten musical symbols using artificial intelligence will be developed. In a further stage, it is planned the development of a database, of a web server and a website. The digitized original documents, its digital component saved as MusicXML and its meta-data inserted by the user will be stored in the repository. The web server will be responsible for the processing of research modules and of the optical recognition of the scores. It will also be responsible of making the interface between the

[1] http://imslp.org/

repository and the website. Finally, the website will be in charge for the interface with the user and for the managing of all of the system. (Figure 1.1 *in [2]*). The digitized original documents, its digital component saved as MusicXML and its meta-data inserted by the user will be stored in the repository. The web server will be responsible for the processing of research modules and of the optical recognition of the scores. It will also be responsible of making the interface between the repository and the website. Finally, the website will be in charge for the interface with the user and for the managing of all of the system. (Figure 1.1).

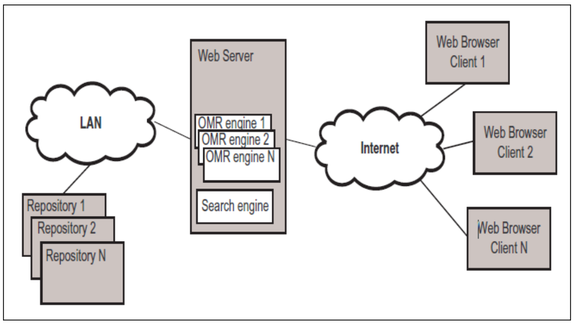


Figure 1.1: System Architecture *in [2]*.

The developed work along this dissertation focus on a specific step of the Web Server block (Figure 1.1), more specifically on the Optical Music Recognition (OMR) engine. This module is the responsible for all the digitized music scores interpretation, which means that it receives as input an image and have as output a digital file with all the relevant information in a machine readable format. Considering that the music relevant information remains basically on the type and relative position of the symbols, the logical step is to segment the symbols in order to identify them. To improve the symbol segmentation and obtain a clear and more accurate identification, the stafflines should be removed. All the work described on this essay is based on this point, the staffline detection and consequent removal.

1.2 Objectives

The main objectives of this dissertation are: review and compare the state of the art techniques used on the staffline detection and removal; research and develop a method to detect and remove stafflines from a music score in the grayscale domain; evaluate the developed algorithm performance and compare the obtained results with other state-of-art methods.

1.3 Contributions of this project

The work developed in this dissertation brought some innovation to the OMR systems introducing a method that avoids the binarization step and therefore eliminates the errors inherent to this stage. Throughout this work the paper “Staffline Detection on grayscale images” were submited and accepted in the 1st PhD. Students Conference in Electrical and Computer Engineering[2] (StudECE) (Appendix A) and one article is being prepared for submission to the 6th Iberian Conference on Pattern Recognition and Image Analysis[3] (ibPRIA).

1.4 Dissertation Structure

This dissertation is organized in six chapters aiming to report all the steps developed in this project. On the first chapter, the main objective is to contextualize the work and to show some motivations. On Background Knowledge (Chapter 2) is presented and explained some methods and technologies that are fundamental to the understanding of the next chapters and the project itself. In the next chapter, the State-of-art (Chapter 3), some current methods developed in the field of the OMR systems, including the methods under the recent staff removal competition within the International Conference on Document Analysis and Recognition (ICDAR2011) are studied. The Chapter 4, Staffline Detection and Removal, is where all the developed code and methods are explained and it is followed by Chapter 5, where the methods used to check the performance are explained and the final results are disclosed and criticized. At last, in the Conclusion and future work chapter (Chapter 6), some final comments and last conclusions are made as well as some proposals to improve the current method in the future.

[2] http://paginas.fe.up.pt/ StudECE2012/
[3] http://www.ibpria.org/2013/

Chapter 2

Background Knowledge

2.1 Technologies

In this project, three environments were used: *Python*, *Matlab* and *VisualStudio*. The choice of this combination of applications aims to usufruct the best from each of them. All of this work could, for example, be done with the use of only one of these programs, but the complexity would be unnecessarily higher. For these reasons, *Python* (cf. 2.1.1) was used for the user interface, due to the easiness that this gives when dealing with the visualization and graphic manipulation. *Matlab* (cf. 2.1.2), beyond its user-friendly interface, its huge functions library oriented to operations on matrices, make it an essential tool when it is needed the mathematics manipulation of an image. Regarding *VisualStudio* (cf. 2.1.3), it is one of the best applications to develop projects in C/C++ for Windows. This language is very versatile and due to its low level, allows the elaboration of programs in a quite efficient way, controlling its functioning with precision.

2.1.1 *Python*

Python[1] is a dynamic programming language. This language is used in various application domains and it is characterized for its versatility and simplicity since it possesses various libraries which can be integrated on other programming languages like the libraries *Jython* for *Java* or *IronPython* for .NET. Concerning this essay it was used as an interface for the creation of a database used as point of reference to verify the efficiency of the developed program.

2.1.2 *Matlab*

Matlab[2] is a language of high level with an interactive environment, which allows developing complex computational tasks, in a faster way than the traditional languages programming such as C and C++. This software was used to produce and study some results, in order to check the method efficiency.

[1] http://python.org/
[2] http://www.mathworks.com/products/matlab/description1.html

2.1.3 *VisualStudio*

VisualStudio[3] is a free platform with various applications of development to create web applications, from a client (Windows) and for a *Windows Phone 7*. In this case this platform worked as a compiler of C/C++, language which was used to develop the core application of this project.

2.2 Methodology

There are some concepts that are crucial to the understanding and development of this work. This section outlines some of them.

2.2.1 Graph

A graph is a pair $G = (V,E)$ of sets such as $E \subseteq [V]^2$([4]), so, the elements of E are subsets of two elements from V. The elements from V are vertices or nodes from the graph G, while the elements of E are its edges [6]. A graph with a set of vertices in V is denoted as *graph in V*. The order of a graph G is represented for a $|G|$ and it is given by the number of vertices that this possesses, the number of edges is represented by $||G||$. If v belongs to e, then e is an edge of v. If $x \in X$ and $y \in Y$, then xy is an edge between the nodes X and Y and belongs to $E(X,Y)$ which is the set of all the edges that connects X to Y. All the edges that belong to an apex v are referenced as $E(v)$. Two vertices x,y from a graph $G = (V,E)$ are adjacent if $xy \in E$. Regarding the edges, they are adjacent if they have at least one node in common. A graph, depending on the application, can be attributed to its edges a weight and/or a direction, in this last case, the graph is denominated digraph. In the graph theory there are some algorithms (e.g. dynamic programming) which allows solving some more generic issues such as, for example, the issues from minimum path that aim to find the course with the shortest route between two nodes.

2.2.2 Dynamic programming

The dynamic programming is associated to optimization methods to the problem solving that require interrelation sequential decisions that can or cannot be affected by costs that concern future decisions.

One of the characteristics of a directed acyclic graph (DAG) [3] is the ability of its nodes to be linearized, that is, that could be reorganized in a such a way they are disposed in a line and with the edges oriented from the left to the right (Figure 2.1).

If considered the distances to the S node, more specifically from the D node, in a way to demonstrate the reason for this reorganization in the calculation of the minimum path. The only way to reach the D node is through the B and C nodes. Therefore, to reach the minimum path to D, it is necessary to compare these two routes:

[3] http://www.microsoft.com/visualstudio/en-us/products/2010-editions/express

[4] V^2 is the group of all the non-ordered pairs of V elements

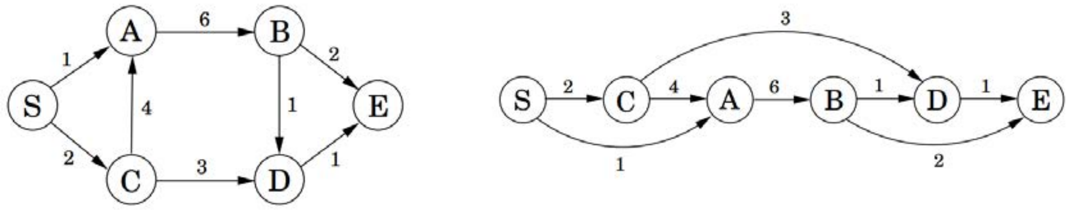


Figure 2.1: Example of the linearization of a graph *in [3].*

$$dist(D) = min\{dist(B) + 1, dist(C) + 3\}$$

.

All nodes can be referred in a similar way. If all values *dist* from the nodes from figure 2.1 from left to right are processed, we can assure that any time a new node is reached, for example node v, we already possess all the necessary information to proceed to the calculation of its distance to the initial node. Because of that we conclude that all values $dist(.)$ can be calculated all at the same time.

Initialize all the values $dist(.)$ like ∞
$dist(s) = 0$
For each $v \in V \backslash \{s\}$, in the linear mode:
    $dist(v) = \min_{(u,v) \in E}\{dist(u) + l(u,v)\}$

The algorithm above solves the problem step-by-step, which results in a simpler solution. The easiest way to solve, is the calculation of the value of $dist(s)$ which easily can de concluded as nought. Afterwards the values of the adjacent nodes are calculated.

2.3 Image Binarization Procedures

In most OMR systems, the first step is the binarization step. Various algorithms where developed in the past in a way to make this process automatic and efficient, but its result always depends on the specific case to be analysed. The binarization process is rarely target of a big analysis from the investigators, which usually prefer to use generalist binarization, processes such as the Otsu method [7]. This method which is considered one of the best and fastest global thresholding method [8], considers that the image has two categories: background and foreground and computes a threshold that maximizes the number of elements of both categories.

On other articles such as [9, 10] it is used the Ridler and Calvard automatic threshold [11] technique. The binarization methods can calculate globally or locally the threshold. Using the threshold calculated globally allows the extraction of objects in a quite quick and efficient way from images with a uniform background, but when it has lots of noise or suffers some variations on its background structure the use of the local threshold calculation is more efficient, besides the heavier computational cost.

Before BLIST, all the binarization methods were generic for all kind of images ignoring their content [12]. It is evident that the development of specific methods of score binarization, allows obtaining more reliable results, and consequently, to improve the remaining steps of the OMR process [13]. The BLIST method emerges as an alternative binarization method to improve the obtained results in the first steps of the OMR systems, the staffline detection. The foundation of BLIST is to maximize the quality of binarized lines by maximizing the number of the pairs of consecutive runs summing line thickness+spacing [12] (cf. section 4.1.1). This runs consists on computing the histogram of every possible binary image, which means using every possible thresholds to binarize the grayscale image (same method used in section 4.1.1).

Chapter 3

State of the art

3.1 Optical Music Recognition - OMR

Most of the authors in literature related to the OMR systems agrees that the best approach to implement these systems is through the fragmentation of the problem in smaller steps [13]. An example of this modular fragmentation, is the model found in [5]. In this article the reading of the score is bases in four main steps:

1. Pre-processing;
2. Detection of music symbols;
3. Rebuilt of the music information in a digital format;
4. Creation of a model musical notation so that, using the digital format, it symbolically represents the score.

There are various methods to solve each of the described above modules. During the pre-processing, several techniques such as improvements, binarizations, noise removal, deskewing, among others (cf. chapter 3.3) could be applied. In this module is also gathered informations about the line spacing and the thickness of the same stave, in a way to create references for future comparisons (cf. chapter 4.1.1).

The detection of music symbols is sub-divided in three sub-modules: the detection and stave removal (cf. chapter 3.4) in a way to obtain just the music symbols; segmentation of symbols and, in the end, recognition of symbols.

The second and third modules appear sometimes combined. In this step, beyond the graphic analysis, a reconstruction of the musical information in a logical description is made.

At last, on the final module, the information obtained in the previous steps is analysed, and subsequently compiled in a final file of MIDI or MusicXML type. v Despite this being one of the most frequent structures for the OMR systems architecture, other models were proposed to improve the obtained results (e.g. some models compare several methods in order to pick the one that fits better a certain image and that have better results [14, 4]).

The methods used in [4] can be described in the following manner (Figure 3.1): a score is digitalized and converted into MusicXML using different OMR programs; the results are normalized and saved on the correspondent step; subsequently, the results of each step are compared and rules are applied that combine all versions in a way to minimize the applied errors. The core of this procedure are the rules to combine the versions. This requires the manual creation of a map with the advantages and disadvantages of all the steps of each program, such as the one found in [14].

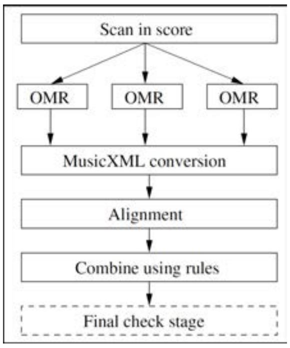


Figure 3.1: Procedure for the digitalization of a document in *in [4]*.

Other works combine, for example, the information obtained through the OMR methods, with audio information, to solve ambiguities [15, 16].

3.2 Characterization of the intervention domain

The detection of the stafflines of a score is the first step of most OMR processes. Its great relevance comes from the easiness how posteriorly we can execute the extraction of the musical symbols. The obtaining of staves is usually done using the binarization of the images through the definition of thresholds that can be local or global. Yet, these techniques can withdraw some relevant information of the stave and introduce noise on the further steps of the process. So, appears the need to create a method that allows eliminating the loss of information resulting from the binarization, a way to obtain the wanted characteristics directly from the image in grayscale.

3.3 Pre-processing of images

On what concerns the quality of the digitization, there is no study that analyzes the impact of this step on the global performance of the algorithms, so there is no pattern on the choice of the resolution done by the authors. If, on the one hand, they used resolutions of three hundred dots per inch (300dpi) (as found for example in [1, 17]), on the other hand in others, like [18] and [19], they used, namely, four hundred dots per inch (400dpi) and six hundred dots per inch (600dpi).

Since normally the big issues found are mostly associated to noises, the unwanted variations of luminance and even the bad quality of the document that is being digitalized, the increase of the number of pixels for inch will not contribute to a better performance in further steps. There are nonetheless more efficient ways of working the images such as:

- Quality treatment of the initial image[20]: for example the equalization of the histogram allows to obtain better results during the binarization;
- Binarization [20, 21, 22, 1] (cf. chapter 2.3);
- Noise removal [21, 20]: low-pass filters are applied to reduce “salt and pepper” effect which can further difficult the detection of relevant information;
- Blurring [20]: is applied a Gaussian filter to soften the image;
- Deskewing [20, 21, 22]: sometimes the documents are not perfectly aligned. When this effect is compensated and the documents are aligned, the stave detection from the compass bar and from the musical symbols stems becomes easier.
- Morphologic operation [20]: these operations aim to reduce even more the noise, removing all the sets of 4*4 pixels. Furthermore, the zones that are to the left and to the right of the staves are erased or marked as a zone with no useful information.

3.4 Staffline detection

Since the detection and identification of the musical symbols are conditioned by the efficiency of the detection and stave identification, the development and quality of the final results of all projects depend in great part of this step success. Usually the detection of staves appears related to its removal, but there are some exceptions [23, 24, 18].

One of the methods for detection consists in searching horizontally sections where the number of black pixels is, at most, identified a line each time it finds a local maximum [25, 21, 26, 27, 28, 29]. Yet, this method as a big disadvantage which is to only be able to detect lines that are exactly horizontal, which cannot happen due to small rotations of the score. To rectify this fault, researches can be made using several rotations and consider the rotation that maximizes the maximum number of black pixels found in each line.

Other form of finding it is using vertical lines of research [30]. This way, the lines are travelled in form to detect potential stave lines; this detection is made analysing sections that satisfy pre-defined patterns such as Aspect ratio, curvature and continuity. There are also some methods that improve the previous methods using projections both horizontal and vertical of the symbols and the staves [31, 28, 32].

As referred on chapter 3.3, works such as the Fujinaga [21] use image processing techniques such as the run-length encode (RLE), connected components and projections. The first step of this method is to determine the staffline height and the staffspace height (cf. chapter 4.1.1), using the RLE. Afterwards, the stave position as to be determined before the stafflines can be removed. For that, these ones are isolated from the symbols following two steps. First all the vertical black lines that have more than the double of the staffline weight are removed; then the connectivity among components is analysed and all that have a smaller width than the staffspace height are removed.

There are other techniques to detect staves, such as for example to group the columns having in account its spacing, width and vertical positioning relatively to the image [33], classifying the horizontal segments in rules [34] or doing the tracking of lines [35, 36]. On the article [37], a line segment is composed by several connected vertical black pixels with similar sizes, validated by neighbouring properties.

In spite of the variety of available methods, there are none without any limitation. One of the tougher situations to solve is the existence of lines with discontinuations and that are not perfectly flat. One of the few works that fulfilled some effort trying to solve the discontinuities was [38]. This algorithm does a pixel by pixel research along all the image, classifying it as a spot or as a trace and then tries to build lines that allow to connect the traces.

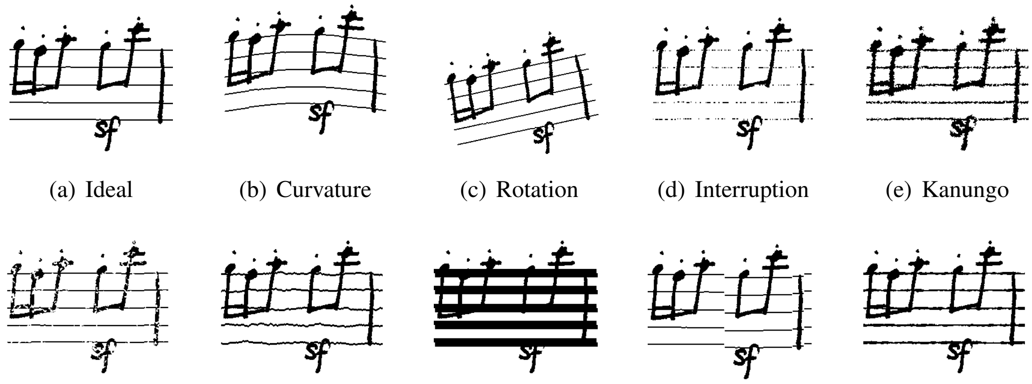
A common point in all of the analysed methods is the fact that the staves need to be binarized before being worked on. This brings the inconvenient of the definition of a threshold that takes a value that is functional to the entire document, in a way to separate the objects under analysis from the score's background. Deteriorated music documents may not have a value for which this condition is verified so, after the binarization the score may have zones where the data will be lost (Figure 3.2). This dissertation will elaborate an algorithm that allows the score detection with no use of binarization, working in grayscale images.

Figure 3.2: Result of a binarization error (figure 2b in [5]).

One of the few articles that reports the use of grayscale images is the Robust Staffline Thickness and Distance Estimation in Binary and Gray-Level Music Scores[5]. In this article, an analysis method on an image with grey levels is used to obtain the staffline height and the staff space height. For this method, instead of calculating a histogram for a binarized score with a threshold value, the original score is binarized several times using various values of threshold. This way it is possible to work with all values obtained and analyse its tendency, achieving an expected value for this reference measures closer from real and more robust in relation, for example, to the gradients of luminosity and to the paper quality.

3.5 Other OMR systems

The International Conference on Document Analysis and Recognition (ICDAR) [39] promoted in 2011 a competition where several authors could compare the results of their works in OMR systems. It consisted in two different competitions: writer identification and staff removal. This dissertation will give focus on the staff removal. The used database (CVC-MUSCIMA[1]) were formed by 1000 handwritten music scores. As a way to test the efficiency of the different methods, several distortions were applied to these images, such as: degradation with Kanungo noise, rotation, curvature, staffline interruption, typeset emulation, staffline y-variation, staffline thickness ratio, staffline thickness variation and white speckles (Figure 3.3).



(a) Ideal (b) Curvature (c) Rotation (d) Interruption (e) Kanungo

(f) White speckles (g) Staffline y-variation (h) Staffline thickness Ratio (i) Typeset emulation (j) Staffline thickness Ratio

Figure 3.3: Applied distortions.

The methods used in this competition will now be briefly described.

From the Computer Vision and Pattern Recognition Unit of the Indian Statistical Institute, Kolkata, India, the authors Jit Ray Chowdhury and Umapada Pal used a system that thins the original image and categorize the thinned parts in two groups: images containing straight lines and images with curved lines. The first group is then subdivided in horizontal and non-horizontal lines. After these steps, the algorithm detects the stafflines using a method described by them as similar to pass a ring on the staffline. If it finds obstacles like music scores, that portion is removed based on some measures. To detect the stafflines, the authors compute the staffline height, the staff space distance, the vertical positional variance of the pixels of thinned lines, etc.

Other method was developed by Bolan Su, from the School of Computing of the National University of Singapore; Shijian Lu from the Institute for Infocomm Research, Singapure; Umapada Pal from the Computer Vision and Pattern Recognition Unit of the Indian Statistical Institute, Kolkata, India; and Chew-Lim Tan from the School of Computing of the National University of Singapore. This method starts using a histogram of vertical run length to calculate the staff height and staff space. Considering the stafflines parallel, the staff height and staff space are used to estimate the stafflines direction and to select a staffline curve that suits better each image. This staffline curve is then used to detect the true stafflines on the picture and remove them.

[1] http://www.cvc.uab.es/cvcmuscima/index$_{database.html}$

Three other methods were submitted by Christoph Dalitz and Andreas Kitzig from the Niederrhein University of Applied Sciences, Institute for Pattern Recognition (iPattern), Krefeld, Germany. The first one considers long horizontal runs as candidates. In order to work with music scores with some small rotation or curvature, this method uses a preprocessing step that rotates the image, aligning vertical strips based on their projection correlation [21].

The second method computes the staffheight detecting the most frequent vertical run length. The detection of the stafflines is done by vertically thinning long horizontal runs with an average black percentage above a threshold (this step limits the detection of curved stafflines) and then, after grouping this horizontal lines based on the staff height and distance, the staffs are connected as sub-graphs. At last, the staff height is used to remove all the vertical runs with less than $2 * staffheight$ around the stafflines.

The third method submitted by this group of authors is based on the construction of a graph comprising the branching and the corner points of the skeleton image. Using heuristic rules, the graph vertical and horizontal links are created giving evidencing the stafflines. Since the horizontal linking is based on extrapolation, the method fails for heavily curved stafflines.

The stable path method also participated in the competition, but it is studied on Chapter 4.

All of these methods were developed to work with binary image, since the proposed database itself comprises only binary images. Besides the stable path method, that is upgraded in this work to work with grayscale images, there are two methods that could be adapted to work in grayscale too, the first and second methods developed by Christoph Dalitz and Andreas Kitzig. The first method, starts by considering long horizontal runs and then rotates the image in order to align vertical strips based on their projection correlation. In order to work in grayscale domain, this method could do a similar approach to the one used on Robust Staffline Thickness and Distance Estimation in Gray-Level Music Scores [5]. That means, apply the existing method to all the binary images generated by all the possible thresholds and then instead of searching the long horizontal lines, it could detect small horizontal segments and later use them to check if there are some combinations of small segments that form a long horizontal line. Once the candidates are detected the second part of the method can work in the same way.

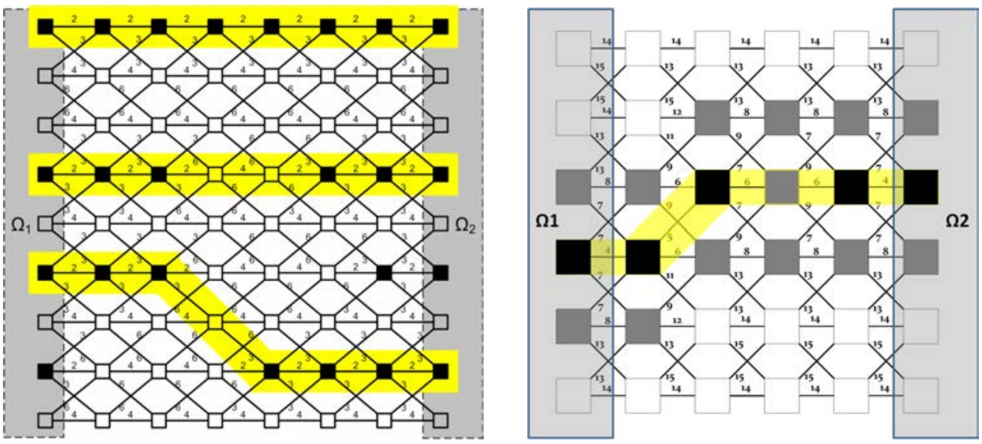
The second method could also be adapted to work on the grayscale domain using an approach similar to the first method as regards the use of the binary images generated by all the threshold, but instead of searching for long horizontal lines it as to search for small vertical lines that meet the conditions defined in the binary mode. In a certain way, this method is similar to the one used on the Staff Pixel Detection algorithm (4.1.2) but with different rules.

Chapter 4

Staffline Detection and removal

The main concept of the method developed along this dissertation is the same described on [1] and it is based on the stable path algorithm. In order to use this method, the image is considered to be a graph (Figure 4.1) where the value of the pixels represent the nodes and the edges connecting neighbouring pixels, represents the cost w. This weight w is obtained using the weight calculation algorithm (cf. section 4.1.3) that uses the value of each pixel, as well as their relative positions and the luminance value of their neighbours.

The path between a pixel u and a pixel v is defined by a list of all the pixels it passes by and their cost is the sum of all the weights between each pair of pixels so, at this point, the shortest path between the pixels u and v is the one with the lowest cost. Assuming that a staffline do not zigzag from left to right, the computation of the path can have its search restricted to paths with only one element per column, which means that the optimal staffline can be found using Dynamic programming 2.2.2. The use of only one element per column implies a 45° limit rotation of the stafflines.



(a) in binary images　　(b) in grayscale images

Figure 4.1: Shortest Path graph.

The Figure 4.1(a), represent a small model graph of a binary musical score with four stafflines in rows 1, 4, 6 and 8, composed mostly by black pixels but with some white pixels that represent some noise. Although this method works on both binary (Figure 4.1(a)) and grayscale (Figure 4.1(b)), the weights are calculated using two different weight algorithms (cf. 4.1.3).

At this point the most important thing to retain is that the shortest path algorithm chooses the darkest path. So, analysing one path at a time, in Figure 4.1(a) the shortest path between Ω_1 and Ω_2 is the first row since it is composed only by black pixels. Using this method, the four paths could be sequentially detected in four iterations. Since all the paths have some common characteristics which makes them a potential staffline, the stable path concept uses that characteristics to find all the paths simultaneously. Using the grayscale weight method (cf. section 4.1.3) on Figure 4.1(b) the path will be the one following the darkest pixels, in this case the black ones.

Definition. *A path $P_{s;t}$ is a stable path between regions Ω_1 and Ω_2 if $P_{s;t}$ is the shortest path between $s \in \Omega_1$ and the whole region Ω_2, and $P_{s;t}$ is the shortest path between $t \in \Omega_2$ and the whole region Ω_1. Quote from [1].*

The result of this method can be checked in Figure 4.2. The tendency for the paths to follow the stafflines are evident in both directions.

(a) Stable Paths from right margin to left (b) Stable Paths from left margin to right

Figure 4.2: Illustration of Stable Paths.

A stable path is considered a staffline if, after this method is applied in both directions as described above, the two endpoints of a direct and reverse path coincide. The Listing 4.1 explains the previous algorithm using high level function.

4.1 Preprocessing

4.1.1 Staffdistance and Staffheight

The proposed algorithm starts with the detection of the staffdistance and staffheight. To do so, it uses the Run Length Encoding (RLE) algorithm to represent the pixels along a column of a binarized image. So, besides the inherent compression, it also saves the number of times that each black or white value repeats, e.g. the sequence 1 0 0 1 1 0 1 1 1 0 0 1 1 1 1 is encoded as 1,2,2,1,3,2,4. From this result we can get the white pixels in the even positions and the black in odd positions, given that it always start with the number of white pixels. To find the estimated value of staffheight, the most frequent value in odd positions is analysed and to get the value of Staffdistance, the even positions are analysed for the most frequent value too. To work with

Listing 4.1: Program algorithm (Based on previous work described in [1]).

```
Preprocessing:
        compute staffdistance and staffheight
        compute staff values
        compute weights of the graph
Main Cycle:
        compute stable paths
        validate paths with blackness and shape
        erase valid paths from image
        add valid paths to list of stafflines
        end of cycle if no valid path was found
Postprocessing:
        delete stafflines with large variations
        uncross stafflines
        organize stafflines in staves
        smooth stafflines
Staff Removal
```

grayscale images, instead of computing the histogram of the runs for a single binarized score, the histogram of the runs for "every" possible binary image by varying the threshold from a low to a high limit is computed.

Figure 4.3: Staffdistance and Staffheight representation.

4.1.2 Staff Pixel Detection algorithm

The Staff Pixel Detection algorithm returns three values referent to the minimum, median and maximum grayscale level of the pixels that comprise the stafflines. This function uses the previously calculated staffdistance and staffheight in order to verify whether a pixel belongs to a staffline. Each image is binarized using different threshold values (from 1 to 90% of the maximum grayscale range), and for each level, the image is scrutinized in order to find sets of pixels that fulfils the conditions to belong to a staffline: have a quantity of pixels with values below a threshold equal to the staffheight and a quantity of pixels with values above threshold equal to the distance between staffs (Figure 4.4).

If the pixel validates these conditions it is stored. For each threshold, the minimum, median and maximum values are computed and saved on three vectors so, in the last stage, the median of those vectors allows to get the expected results. The use of the median of the three vectors (minimum, median and maximum) allows the results to be more robust to noise. This algorithm

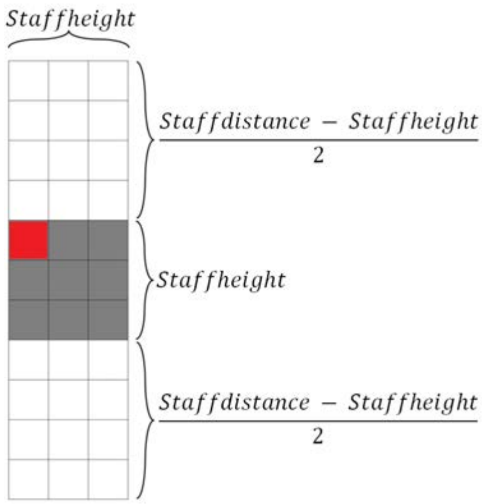


Figure 4.4: Rules to detect if neighbour pixels around the red one belong to staffline.

also marks the used pixels on an image (Figure 4.5) in order to use them in later stages of the program.

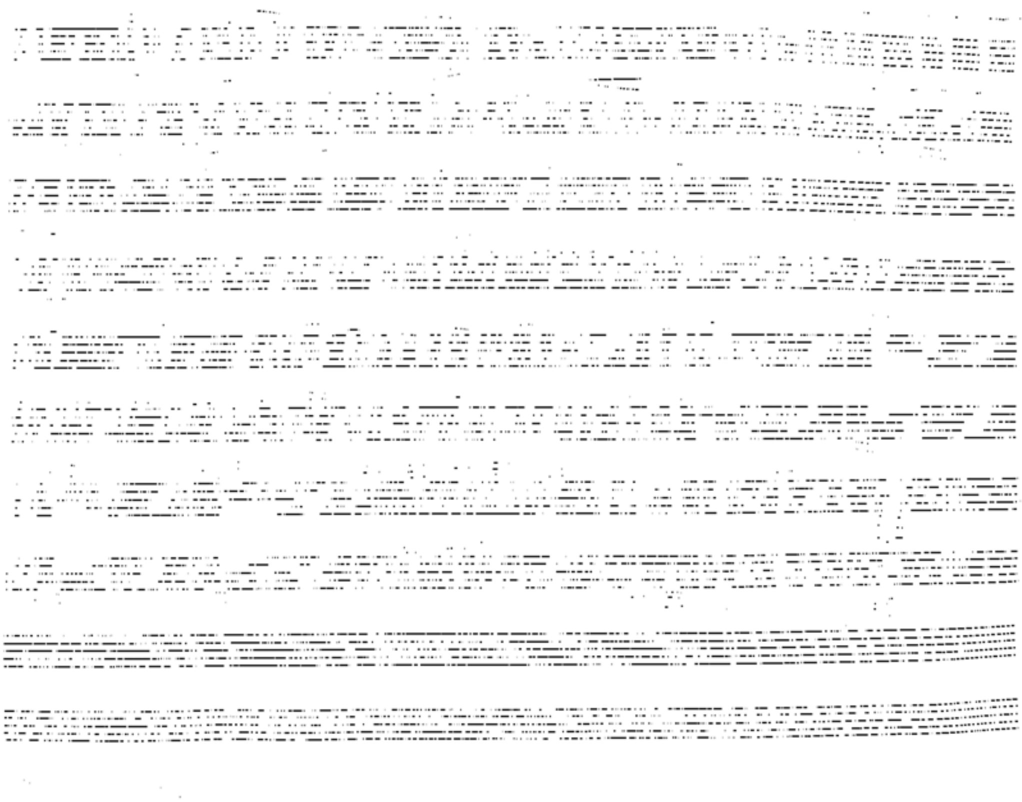


Figure 4.5: Example of an image with the detected staff pixels.

Listing 4.2: Binary Weight algorithm.

```
WeightFunction(pixelValue1, pixelValue2, vRun1,
        vRun2, nearestVRun1, nearestVRun2, NeighbourhoodType)
{
        value = min(pixelValue1, pixelValue2);
        weight = base Weight (value, NeighbourhoodType);
        if((vRun1<=STAFFLINEHEIGHT)
        OR(vRun2<=STAFFLINEHEIGHT))
        weight = weight - delta;
        if((nearestVRun1>=STAFFSPACEHEIGHT+
        STAFFLINEHEIGHT)
        OR(nearestVRun2>=STAFFSPACEHEIGHT+
        STAFFLINEHEIGHT))
        weight = weight + delta;
        return weight;
}
```

4.1.3 Weight calculation algorithm

On previous work for binary images [1], the weight function was developed to work with the values of the incident pixels, assigning a low cost to the edge if any of the pixels are black and a high cost otherwise. Taking in to account that this function is projected to work with music scores, it is wise to use some rules based on prior knowledge about stafflines. In order to include this concept on the code, two more attributes were taken into consideration aiming to penalize black pixels that belong to a symbol and to promote a black pixel that belong to a staffline. If the analysed black pixel is vertically contiguous to other black pixels (vrun[1]) and the height of the set they form do not surpasses the staffheight previously defined, then the pixel is more likely to belong to a staffline than to a symbol, therefore the cost of that edge is benefited by a term in the weight algorithm. Our knowledge of music scores also allows us to notice that, when a line is part of a stave, it is expected that approximately at a staffdistance, there is another line, i.e. there is a staffdistance between each valid vrun. So, if two valid vrun are too far from each other, one of them probably belongs to a symbol like a ligature, and consequently it will be penalized on the weight algorithm. The Listing 4.2, shows the weight algorithm pseudo-code developed in [1] to use with binary images.

The grayscale weight algorithm determines the cost of a certain edge based only on their two edges (value of correspondent pixels). Unlike the weight algorithm used for binary images, it handles different intensity values of pixels. In order to be assigned a specific value to each edge, a weight continuous function (4.1) is defined by branches. Each branch is defined between the minimum, median and maximum values obtained from the staff pixel detection algorithm (cf. section 4.1.2) and the values inside the correspondent intervals are interpolated. Some values were empirically obtained and others are based on the binary method. Considering the function:

[1] vertical run

```
w=interpolate(px,x1,y1,x2,y2,...,xn,yn)
```

where w is the return value correspondent to the weight of the pixel, px is the pixel grayscale level and $x1$, $y1$, $x2$, $y2$, ... , xn, yn are the points to interpolate. The pixel_weight function can be defined as:

$$pixel\_weight(pixel) = \begin{cases} interpolate(pixel, 0, 5.5, minimum, 5.0) \\ \quad \text{if } pixel >= 0 \wedge pixel < minimum \\ interpolate(pixel, minimum, 5.0, median, 4, maximum, 5.0) \\ \quad \text{if } pixel >= minimum \wedge pixel < maximum \\ interpolate(pixel, maximum, 5.0, 255, 8) \\ \quad \text{if } pixel >= 0 \wedge pixel > maximum \ \wedge pixel < 255 \end{cases} \tag{4.1}$$

A toy example, illustrated in Figure 4.6, was used to represent a graphic of the weight function.

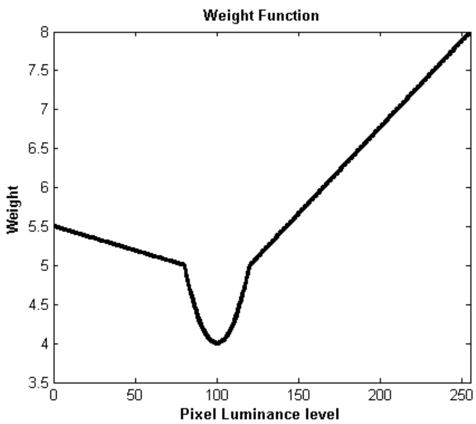


Figure 4.6: Weight Function example.

As well as the binary method, there are some rules that can be applied to this function based on the staffline background knowledge, but since the staff pixel detection algorithm (cf. section 4.1.2) already found the position of some pixels that are part of the stafflines, this information will now be used again in order to penalise the pixels that are not part of a staffline. The Listing 4.3 represents the grayscale weight function pseudo-code.

4.2 Main Cycle

On the main cycle, the program use the previously created graph in order to find the stable paths between both margins of the music score. To be considered a staffline, the detected paths must pass two criteria: have a percentage of dark pixels above a predefined threshold and have a shape similar to the path with average darkness. If the path fails this two parameters it is disposed. The term dissimilarity was developed on the program and was used as threshold to check to paths have similar shapes. It was defined as the difference between two paths (disregarding the offset).

Listing 4.3: Grayscale Weight algorithm.

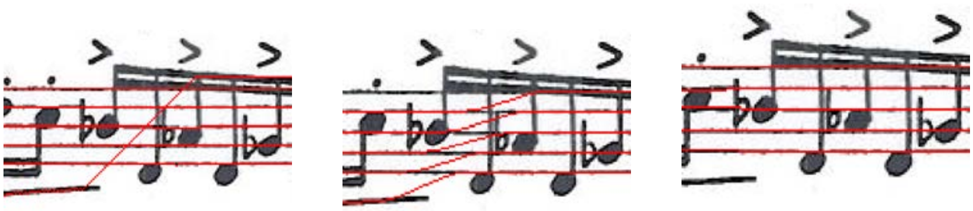
```
WeightFunction(pixel1, pixel2, NeighbourhoodType,
        reference_image, minimum, median, maximum)
{
        value = (pixel1 / pixel2)/2;
        weight = interpolate (value, minimum, median, maximum)
        penalization = 0;
        if(NeighbourhoodType=NEIGHBOUR8)        penalization++
        if(pixel1 \notin reference_image) penalization++
        if(pixel2 \notin reference_image) penalization++
        if(penalization > 0) weight= weight * penalization
        return weight;
}
```

Its considered that two paths are to dissimilar if this term is higher than four times the distance between stafflines.

During this cycle, in order to prevent multiple detections of the same staffline, every time a staff is detected it is deleted, which means, set path edges weight to the highest value. This operation ends when there are no more available valid paths.

4.3 Postprocessing

Although at this point, all the stafflines are detected, due to some occasional problems during the previous method, there is a need to rearrange the lines based on the knowledge we have about stafflines. On the developed algorithm there are five methods responsible for ensuring compliance with these rules. Sometimes, during the detection, some found paths that have a strange behaviour in comparison to the others and cross several other lines (Figure 4.7(a)). If this false path is not removed, this error will spread to the rest of the postprocessing function and, on the final results, instead of just detect one false staffline, the program will detect five false stafflines and five missed (Figure 4.7(b)).



(a) Problem (b) Results without removal step (c) Results with removal step

Figure 4.7: Multiple cross path.

To solve this problem, an algorithm was implemented to remove lines with this strange behaviour and applied in the beginning of the postprocess. This algorithm starts by finding all the intersections of the stafflines and storing their corresponding references (i.e. if staffline 2 crosses staffline 3, it will be stored a pair like $(2,3)$). Then, considering that it is not an error if two stafflines cross several times with each other (Figure 4.8), the repeated pairs are replaced by a single reference. At last, it is detected if a staffline crosses more than three (corresponds to cross half of a stave) consecutive stafflines and, in that case, it is removed (Figure 4.7(c)).

Figure 4.8: Pair of crossed stafflines.

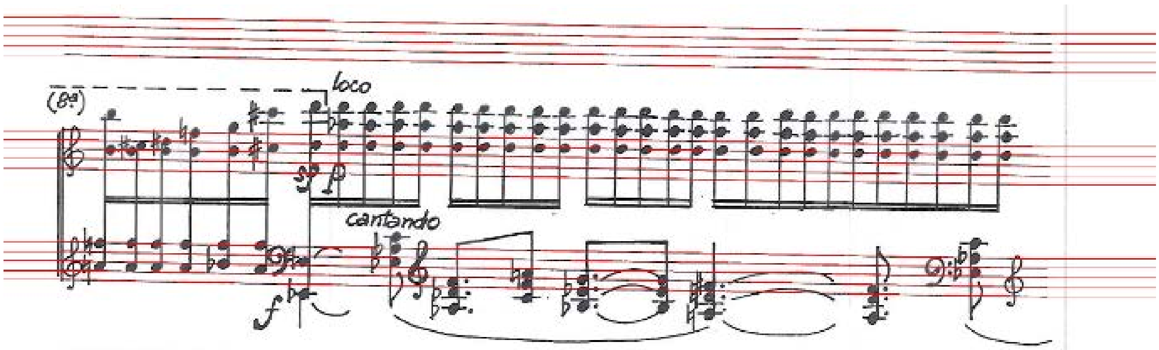
The next step is responsible for uncrossing the stafflines (Figure 4.8). Due to some discontinuations of the line, to some noisy regions or even due to some symbols, two lines can be detected swapping positions. This function solve this by checking each column of the musical score for stafflines with a common point and, every time it happens, the remaining points of each staffline switch position. For the result of this method to be perceived, the Figure 4.9 is already affected by smoothing.

Figure 4.9: Uncrossed stafflines.

With the lines uncrossed, it is now possible to remove spurious stafflines and reorganize the stafflines in sets. This new function iterates through all the lines and whenever the distance between two lines is higher than defined threshold (two time the distance between staffs) it starts a new set. Sets with just one staffline are probably a wrong detection and as such they are eliminated (Figure 4.10).

On the last step of the post processing, the lines are trimmed and smoothed (Figure 4.11). Considering that the path starts on an edge of the musical score and ends on the opposite edge, it have to travel through a lighter area of pixels before reaches the staffline and at the end of the staffline. To remove this part of the path that does not belong to the staffline, this operation works per staff, computing the median colours of every column. Then, starting in the middle of the staffline and moving to both sides, a vertical run with size two times the distance between staffs is

(a) Before organizing



(b) After organizing

Figure 4.10: Images before and after set organizing.

tested until a run is to light, these are the points where the trimming will be done. At last, the lines are smoothed using a standard average low-pass filter.

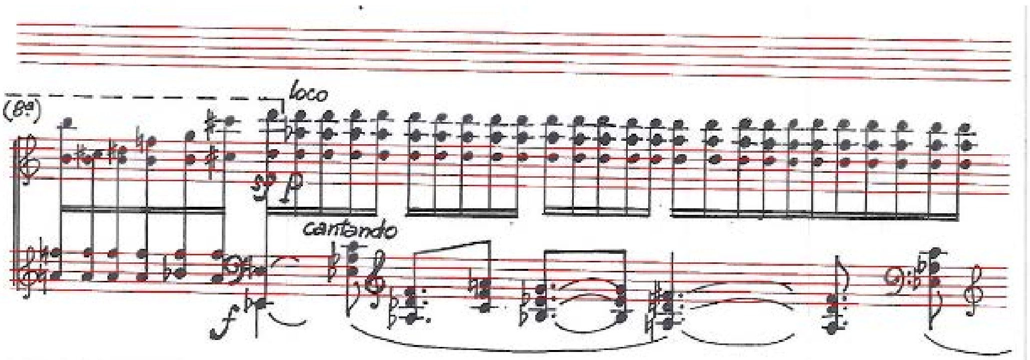


Figure 4.11: Stafflines after trimming and smoothing.

4.4 Staff removal

The final stage of this program is the staff removal. This function was developed considering two parameters, the staff height (obtained with function 4.1.1) and the maximum value of the staffline (obtained with 4.1.2). This operation starts with a vertical run for each staffline pixel. The pixels value will be checked upward and downward the reference pixel to find the position of the staffline edges.

The edge criteria is when a pixel values surpasses the maximum of a staffline or when the run is bigger than twice the staffheight. With the obtained points the size of the detected vertical run is defined and if it is smaller than twice the staffheight its considered a staffline and therefore it is removed, otherwise it is considered a symbol and the pixels are kept. In Figure 4.12 there is a real example where the pixels inside the left box will be removed and the ones inside the one on the right wont. An example of the final result can be checked on Figure 4.13.

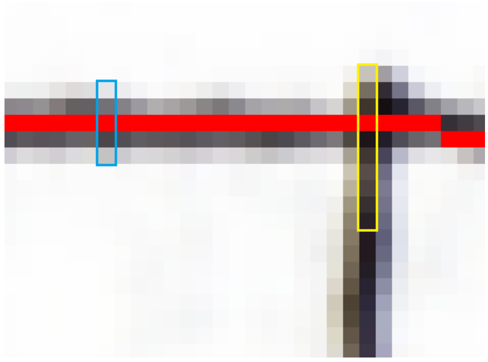


Figure 4.12: Detected staffline.

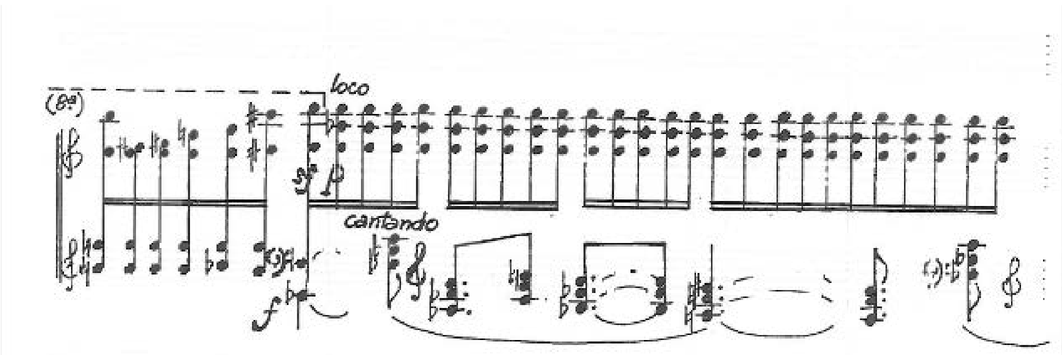


Figure 4.13: Final result – after Stafflines removal.

Chapter 5

Results

5.1 Database

The performance of the algorithm developed in this dissertation was tested using two sets of images (Figure 5.1). The first set, with 64 images, was used as reference to the detection of the stafflines efficiency (i.e obtain the number of missed stafflines and the number of false detected stafflines). For each one of this images, there was a file containing all the manually collected stafflines references. To check the staff removal and the staff pixel detection, a subset of 47 images were used. This subset was used as ground truth, which means the images were manipulated in order to keep only the relevant information for the analysis of the final results.

Figure 5.1: Some database music score examples.

5.2 Metrics and Results

The baseline for this work was the results from the Binarization based in LIne Spacing and Thickness (BLIST) method. In a general approach, the comparison was made using a matlab program that returns the percentage of missed and false stafflines and their corresponding deviations. The results obtained can be checked on Table 5.1. Although the value of false stafflines is roughly the same, the grayscale method misses half of the stafflines missed by BLIST. On Figure 5.2, there is an example of a music score where the binary method (BLIST) have some problems and misses some stafflines and the grayscale method finds all the lines. In this same image, in the lower right corner, it is also perceptible that the grayscale method have more success following the stafflines when compared with the binary method. The reason why this happens, is probably because this is a darker area and therefore a problematic point for the binarization step. Another example of the enhanced stability of the grayscale method when compared to BLIST is shown on Figure 5.3.

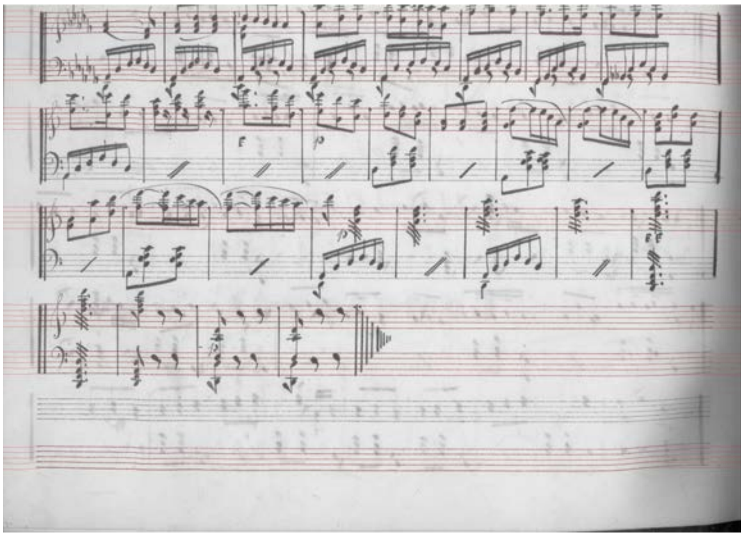
	falseStaff_mean	falseStaff_std	missedStaff_mean	missedStaff_std
BLIST	0,83	2,11	3,57	13,39
Grayscale	0,83	2,29	1,61	7,10

Table 5.1: Global Results (in percentage).

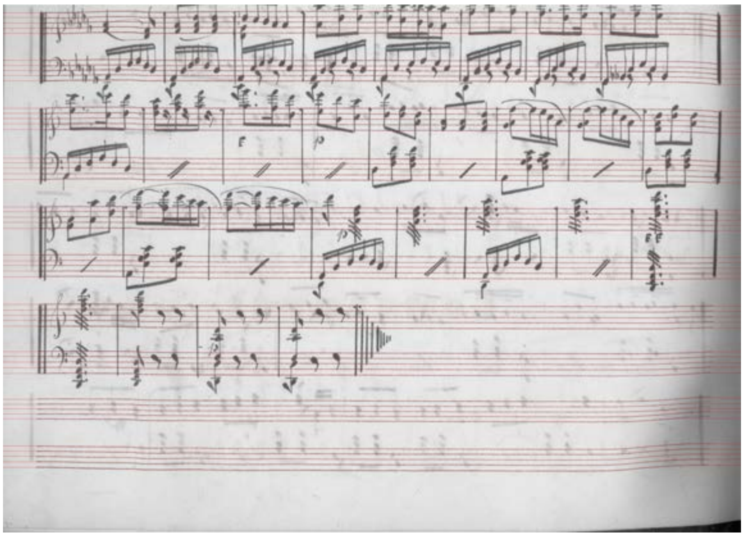
Since this new method was based on the staff pixel detection (cf. section 4.1.2), the performance of this algorithm was tested to assess its reliability. The ground truth images with symbols (Figure 5.4(a)) and without symbols (Figure 5.4(b)) were used as reference.

The image positions of the pixels (Figure 4.5) obtained with the staff pixel detection algorithm were compared with the ground truth images to count the number of false pixels detected and the number of missed pixels. In order to get more accurate results, it should have been used different databases marked by different persons in order to scatter the errors. Since the work is being done using grayscale images, the detected thickness can sometimes be different from the one in binary images. In a way to check if the obtained error in the detection of the staffline pixels was caused by noise or if the algorithm where detecting slightly larger staffs, the ground truth images were eroded in order to enlarge the reference staffline thickness and evaluate the performance in a fairer way. The ground truth images with symbols were used to check if the false positives detected by the algorithm belonged to noise or to symbols. As the symbols grayscale level are similar to the staffline grayscale level, it is far better to detect symbols than noise. In Table 5.2, it is possible to check that the number of detected pixels that are not stafflines nor symbols is near zero, which represents a very satisfactory result. The most important value to consider is the false negative from the Ground Truth without Symbols, that represents the used pixels that do not belong to the staffline. The false positive value checks if there are enough detected pixels to obtain reliable results (this values are usually high, due to the restrict rules imposed by the staff pixel detection algorithm, that gives preference to the quality of the detected pixels instead of the quantity).

It is worth mentioning that the big deviation derives from a specific set of images (Figure 5.5) where the results were not so good (i.e. all the other sets of images had an average false negative

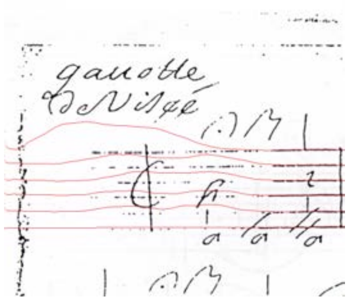


(a) Using BLIST

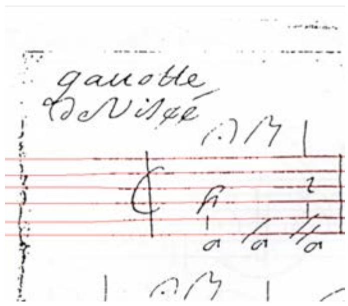


(b) Using grayscale image

Figure 5.2: Final Results.



(a) Using BLIST



(b) Using grayscale image

Figure 5.3: Image section example.

(a) Ground truth with symbols (b) Ground truth without symbols

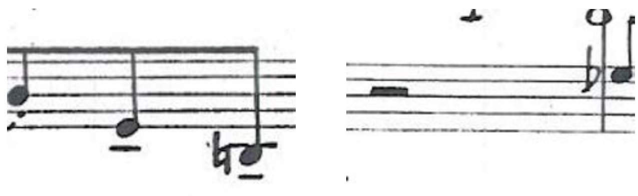
Figure 5.4: Ground truth images example.

error of 2.77% and this specific set has an average of 53.17%). After some analysis, a particular feature of this set was detected. The main problem is the high grayscale level of the pixels that belong to the staffline.

Usually, in an music score, the stafflines are the darkest object, the symbols are a little lighter or similar to staffline and noise have a random grayscale level. The number of pixels that represents noise increase with the threshold. About the stafflines and symbols the order in which they are detected is not very important since usually they are similar. This way, as the threshold increases, the first pixels that meet the algorithm conditions are the ones on stafflines, resulting on a good performance.

The problem with the set of images under review is that the stafflines are the lightest objects in the music score. This means that, as the threshold rises, the noise and symbols are detected and, by the time the threshold is high enough to start detecting the stafflines, there are already to much information detected, making it harder for the algorithm to meet the conditions required to consider that a pixel belongs to a staffline. (cf. Section 4.1.2 in order to understand the Staff Pixel Detection algorithm).

In parallel with this process, the developed Matlab code, compared both the error between the real staffline values (minimum, average and maximum values of the staffline) and the error in the position of the detected pixels. To compute the real staffline values, the code checked the ground truth images black pixels and, in the correspondent position of the grayscale image, the pixel value were stored in a vector used to calculate the real minimum, average and maximum values. The



(a) (b)

Figure 5.5: Example of an image section from the problematic set.

results can be checked on Table 5.4.

The errors presented in this table derives from the same problem discussed before on the detection of the staff pixels position and as proved before, they are not large enough to influence the final results.

As regards the staffline removal, it was used the same method proposed on the International Conference on Document Analysis and Recognition (ICDAR) to test the performance [39]. The proposed formula evaluates the number misclassified staff pixels (misclassified sp) and the number of misclassified non staff pixel (misclassified non sp) in order to the number of all the staff pixels. The error rate (E.R) is then defined by:

$$E.R. = 100 \times \frac{\#misclassified\ sp + \#misclassified\ non\ sp}{\#all\ sp + \#all\ non\ sp}$$

The results are shown on table 5.5.The difference in the quality of both methods is probably because the ground truth images in database are binary, which can result on some minor errors (like on the calculus of minimum, median and maximum value of the staffline).

	Ground Truth without Symbols		Ground Truth with Symbols	
	False Negative	False Positive	False Negative	False Positive
median	3,55	83,53	0	91,44
average	22,08	82,57	0,007	89,70
standard deviation	27,59	13,29	0,02	7,48

Table 5.2: Errors on computed staffline positions.

	miss detected staffline pixels	missed stafflines	false detected staffline
Figure 5.5(a)	35.54%	0	0
Figure 5.5(b)	55.23%	0	0

Table 5.3: Number of false detected stafflines and missed stafflines related with errors in the staff pixel detection algorithm.

	minimum	average	maximum
average error	1%	12%	14%
standard deviation	2%	10%	7%

Table 5.4: Errors on the computation of the staffline minimum, average and maximum values.

	mean error	standard deviation error
BLIST	0.032	0.013
Grayscale	0.042	0.015

Table 5.5: Errors on staffline removal.

Chapter 6

Conclusions and future work

This work main objective was to detect and remove the stafflines using music scores digitized in the grayscale domain. The main idea adjacent to this method was to preserve the information that were otherwise lost in the binarization process (common on OMR systems), making this method a more robust solution to work with music scores with corrupted backgrounds due to problems during the digitization, to damaged paper based documents, among several others anomalies that could be problematic to other systems.

The work in grayscale proved to be quite challenging due to the increased complexity compared to the binary domain, since that, at all times, the computations used all the grayscale levels instead of the two values used in binary images. On the other side, the binary method weakness may also be its strength since it only has to deal with a large amount of noise one time if the binarization succeed with good results, unlike the grayscale method. This was perhaps the most problematic matter.

Overall, the grayscale method had acceptable results, since it brought all the benefits from the previous method removing the errors inherent to the binarization. This means the method still had a good performance when faced with the distortions referred on section 3.5.

Furthermore, the staff pixel algorithm (cf. section 4.1.2) despite being an intermediate module, obtained some satisfactory results and could be used again in other contexts to improve perhaps other OMR systems. The purpose for which it was designed, should be noted when the results are being analysed. Although the number of missed pixels were recorded, they are not very significant since these functions main objective is to detect only staffline pixels and avoid symbols and noise detection, so the rules were the most stringent possible. This way it was assured that the values used to calculate the weight function (4.1) were reliable.

6.1 Future work

On future work, there were some points that could be improved like the post processing (mainly in the construction of the sets and detection of false stafflines) and the penalizations on the weight function. That way the results could be even better since the rules were more restricted. For detection of handwritten music scores, it could also be implemented an OMR system using RGB images. Despite the method would be really complex to implement, the operating principle would be the same as the grayscale method, but instead of using the staffline pixel values it could use a range of colours since, probably, the colour of the stafflines and the colour used to write the symbols are different and more pronounced the difference in grayscale.

Appendix A

Students Conference in Electrical and Computer Engineering (StudECE) – Abstract

Staffline Detection on grayscale images

Vítor C. Vidal, Ana Rebelo, Jaime S. Cardoso
INESC TEC (formerly INESC Porto) and Faculdade de Engenharia, Universidade do Porto

***Abstract*—The detection of stafflines is the first step of most OMR processes. Its great significance derives from the ease with which we can then proceed with the extraction of musical symbols. This step is usually achieved using the binarization of images by setting thresholds that can be local or global. These techniques however, can remove some relevant information of the music sheet and introduce noise in the later stages of the process. It arises therefore a need to create a method that eliminates the loss of information due to the binarization.**
Throughout this paper, the "Staff Detection with Stable Paths"[1] method, that works based on binary images, will be adapted to work with grayscale images. In order to do that, a new weight function that applies a value to each pair of pixels based on the average of their luminance values will be implemented. To improve the results, a function that estimates the range of luminance values of the stafflines will also be created.

I. Introduction

One of the problems that we face today is the deterioration and loss of information over time of unique printed and handwritten manuscripts. Musical scores are among such perishable documents. The digitalization has been used to preserve these documents, but to convert the paper-based music scores into a machine-readable format to facilitate operations such as search, retrieval and analysis, an Optical Music Recognition (OMR) system is needed. In these algorithms, the detection and removal of the stafflines are fundamental to isolate the musical symbols and to have a more efficient and correct detection of each symbol.
Currently, all the OMR methods need to convert the image from grayscale to binary, which results in a possible loss of indispensable information. This work introduces an alternative to all these methods by working over the original grayscale image, which result on more accurate results even on damaged images.

A. Optical Music Recognition (OMR) architecture

According to [2], the most common approach used on OMR systems, and the one used on this work, is to split the problem in four consecutive steps:

- Pre-processing, the stage responsible for the enhancement of the quality of images in order to achieve better results on the next stages using: binarization, noise removal, blurring, deskewing, etc.;
- Symbol detection: subdivided into three sub-steps:
  - staffline detection (section II) and removal;
  - symbol segmentation;
  - symbol recognition;
- Store of the obtained data on a digital file (musicXML, etc.);
- Interpretation of the digital file.

II. Staffline detection

Staffline 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 stafflines lie on the need to isolate the musical symbols for a more efficient and correct detection of each symbol present on the score [3].

A. Compute staffdistance and staffheight (see Fig. 1)

Fig. 1. Staffdistance and Staffheight representation

The method proposed in [3], uses the Run Length Enconding (RLE) algorithm to represent the obtained pixels along a column of a binarized image. So, besides the inherent compression, it also saves the number of times that each black or white value repeats, e.g. the sequence 1 0 0 1 1 0 1 1 1 0 0 1 1 1 1 is encoded as 1,2,2,1,3,2,4. From this result we can get the white pixels in the even positions and the black in odd positions. To find the estimated value of staffheight, the most frequent value in odd positions is analysed and to get the value of Staffdistance, the even positions are analysed for the most frequent value too. To work with grayscale images, instead of computing the histogram of the runs for a single binarized score, the histogram of the runs for "every" possible binary image by varying the threshold from a low to a high limit is computed.

B. Staffline Detection on grayscale images

The usage of dynamic programming and the concept of graphs allowed the creation of the concept of the stable paths [1] to binary images. This concept states that if we seek the shortest path between two edges of a music score and consider that the cost of choosing a path through black pixels is lower than a path through white pixels, all the paths found represent a staffline (Fig. 2).

The main operations of this method – for more information see [1] – include:

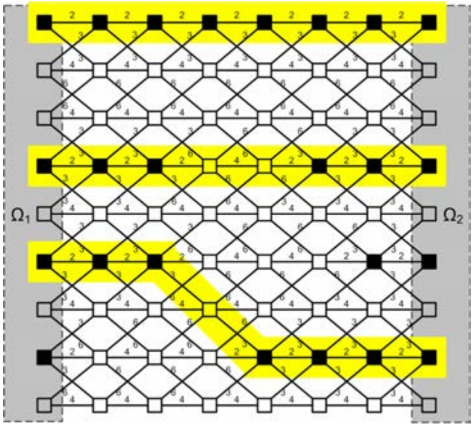


Fig. 2. Example of a path selection in [1]

```
Preprocessing:
  compute staffdistance and staffheight
  compute weights of the graph
Main Cycle:
  compute stable paths
  validate paths with blackness and shape
  erase valid paths from image
  add valid paths to list of stafflines
  end of cycle if no valid path was found
Postprocessing:
  uncross stafflines
  organize stafflines in staves
  smooth and trim stafflines
```

In order to upgrade the previous method to work with grayscale images, a new function to calculate the weight between two pixels is being implemented. This function interpolate the points between three measures: the maximum, minimum and average of the luminance value of the pixels that belongs to a staffline, which are acquired by the function `Staff_characteristics`. This new function uses all the threshold values, previously calculated, that generate the music score staffdistance and staffheight. Then, all the image is scrutinized in order to find sets of pixels that fulfils the conditions to belong to a staffline: have a quantity of pixels with values below threshold equal to the staffheight and a quantity of pixels with values above threshold equal to the staffdistance. If the pixel validates this conditions it is stored and used on the last stage to calculate the expected values (maximum, minimum and average).

The `Staff_characteristics` algorithm:

```
for each threshold,
  vector[threshold] = local_staff_measures
calculate global_staff_measures
use thresholds that creates global_measures
find all sets of pixels like:
  (pixel < threshold) * staff height
  followed by
  (pixel > threshold) * staff distance
calculate the average, the maximum
      and the minimum value of staffline.
```

After the weight of each pixel has being calculated using this function, the average between their two weights is calculated and it is applied to the specific branch. Since the weight function depends on the value of the staffline, we can assure that the path through the stafflines is followed even if there are symbols darker than them. The example in Fig.3 shows a function where the average, minimum and maximum value of the pixels of staffline is correspondingly 120, 100 and 140.

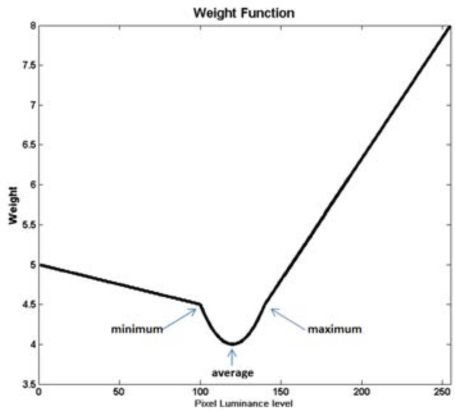


Fig. 3. Example of weight function

III. Future Work

In future work the staffline detection with stable paths [1] will be adapted to the new weight function. The method will successively find the stable paths between the left and right margins, adding the paths found to a list of stafflines, and erasing them from the image. The erase operation is necessary to ensure that a line is not detected multiple times. After the main search step, the stafflines will be post-processed, to assure that the path is valid and respect the previous measurements (staffdistance and staffheight). At this point the paths are also uncrossed, organized in sets, smoothed and trimmed.

Acknowledgment

This work is financed by National Funds through the FCT Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project SFRH/BD/60359/2009.

References

[1] J. S. Cardoso, A. Capela, A. Rebelo, C. Guedes, and J. F. P. da Costa, "Staff detection with stable paths," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 6, pp. 1134–1139, 2009.

[2] A. Rebelo, I. Fujinaga, F. Paszkiewicz, C. Guedes, and J. S. Cardoso, "Optical music recognition - state-of-the-art and open issues," *International Journal of Multimedia Information Retrieval*, 2012, accepted.

[3] J. Cardoso and A. Rebelo, "Robust staffline thickness and distance estimation in binary and gray-level music scores," in *20th International Conference on Pattern Recognition (ICPR)*, August 2010, pp. 1856 –1859.

References

[1] Jaime S. Cardoso, Artur Capela, Ana Rebelo, Carlos Guedes, and Joaquim F. Pinto da Costa. Staff detection with stable paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1134–1139, 2009.

[2] Artur Capela, Jaime S. Cardoso, Ana Rebelo, and Carlos Guedes. Integrated recognition system for music scores. In *Proceedings of the International Computer Music Conference*, 2008.

[3] S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani. *Algorithms*, pages 169–179. McGraw-Hill Higher Education, 2006.

[4] Ian Knopke and Donald Byrd. Towards musicdiff: A foundation for improved optical music recognition using multiple recognizers. In *The International Society for Music Information Retrieval (ISMIR)'07*, pages 123–126, 2007.

[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 (ICPR)*, pages 1856 –1859, August 2010.

[6] Reinhard Diestel. *Graph Theory (Graduate Texts in Mathematics)*. Springer, August 2005.

[7] Nobuyuki Otsu. A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, 1979.

[8] Due Trier and Torfinn Taxt. Evaluation of binarization methods for document images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 17(3):312–315, March 1995.

[9] K. C. Ng and R. D. Boyle. Recognition and reconstruction of primitives in music scores. *Image and Vision Computing*, 14(1):39–46, 1996.

[10] K. C. Ng, D. Cooper, E. Stefani, R. D. Boyle, and N. Bailey. Embracing the Composer: Optical Recognition of Handwritten manuscripts. In *Proceedings of the International Computer Music Conference, Beijing*, 1999.

[11] S Ridler T W Calvard. Picture thresholding using an iterative selection method. *IEEE Transactions On Systems Man And Cybernetics*, 8(Aug):630–632, 1978.

[12] Telmo Pinto, Ana Rebelo, Gilson Giraldi, and Jaime Cardoso. Music score binarization based on domain knowledge. In *Pattern Recognition and Image Analysis*, volume 6669 of *Lecture Notes in Computer Science*, pages 700–708. Springer Berlin / Heidelberg, 2011.

[13] Ana Rebelo, Ichiro Fujinaga, Filipe Paszkiewicz, Andre Marcal, Carlos Guedes, and Jaime Cardoso. Optical music recognition: state-of-the-art and open issues. *International Journal of Multimedia Information Retrieval*, pages 1–18. 10.1007/s13735-012-0004-6.

[14] Donald Byrd and Megan Schindele. Prospects for improving omr with multiple recognizers. In *The International Society for Music Information Retrieval (ISMIR)*, pages 41–46, 2006.

[15] Frank Kurth, Meinard Müller, Christian Fremerey, Yoon ha Chang, and Michael Clausen. Automated synchronization of scanned sheet music with audio recordings. In *Proceedings of the 8th International Conference on Music Information Retrieval (The International Society for Music Information Retrieval (ISMIR))*, pages 261–266, Vienna, Austria, September 2007.

[16] David Damm, Christian Fremerey, Frank Kurth, Meinard Müller, and Michael Clausen. Multimodal presentation and browsing of music. In *Proceedings of the 10th international conference on Multimodal interfaces*, ICMI '08, pages 205–208, New York, NY, USA, 2008. ACM.

[17] A. Rebelo, G. Capela, and Jaime S. Cardoso. Optical recognition of music symbols: A comparative study. *International Journal on Document Analysis and Recognition*, 13(1):19–31, 2010.

[18] Laurent Pugin. Optical music recognition of early typographic prints using hidden markov models. In *The International Society for Music Information Retrieval (ISMIR)*, pages 53–56, 2006.

[19] Ichiro Fujinaga and Jenn Riley. Digital image capture of musical scores. In *The International Society for Music Information Retrieval (ISMIR)*, 2002.

[20] Roland Gocke. Building a system for writer identification on handwritten music scores. In *Proceedings of the IASTED International Conference on Signal Processing, Pattern Recognition, and Applications SSPRA 2003*, pages 250 – 255, Anaheim, USA, 2003.

[21] Ichiro Fujinaga. Staff detection and removal. In Susan George, editor, *Visual Perception of Music Notation: On-Line and Off-Line Recognition*, pages 1–39. Idea Group Inc., 2004.

[22] Alicia Fornés, Josep Lladós, and Gemma Sánchez. Primitive segmentation in old handwritten music scores. In Wenyin Liu and Josep Lladós, editors, *Graphics Recognition. Ten Years Review and Future Perspectives*, volume 3926 of *Lecture Notes in Computer Science*, pages 279–290. Springer Berlin / Heidelberg, 2006.

[23] Hidetoshi Miyao and Masayuki Okamoto. Stave extraction for printed music scores using DP matching. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 8:208–215, 2007.

[24] Mariusz Szwoch. A robust detector for distorted music staves. In *Computer Analysis of Images and Patterns*, pages 701–708. Springer-Verlag, Heidelberg, 2005.

[25] H. Kato and S. Inokuchi. A recognition system for printed piano music using musical knowledge and constraints. In *Proceedings of the International Association for Pattern Recognition Workshop on Syntactic and Structural Pattern Recognition*, pages 231–248, 1990.

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

[27] H. Miyao and Y. Nakano. Note symbol extraction for printed piano scores using neural networks. *IEICE Transactions on Information and Systems*, E79-D:548–554, 1996.

[28] P. Bellini, I. Bruno, and P. Nesi. Optical music sheet segmentation. In *Proceedings of the First International Conference on WEB Delivering of Music (WEDELMUSIC'01)*, pages 183–, Washington, DC, USA, 2001. IEEE Computer Society.

[29] Fubito Toyama, Kenji Shoji, and Juichi Miyamichi. *Symbol Recognition of Printed Piano Scores with Touching Symbols*. IEEE Computer Society, Washington, DC, USA, 2006.

[30] N. P. Carter. *Automatic Recognition of Printed Music in the Context of Electronic Publishing*. PhD thesis, Departments of Physics and Music, 1989.

[31] D. Bainbridge. *Extensible Optical Music Recognition*. PhD thesis, Department of Computer Science, University of Canterbury, Christchurch, NZ, 1997.

[32] Florence Rossant and Isabelle Bloch. Optical music recognition based on a fuzzy modeling of symbol classes and music writing rules. In *Proceedings of the International Conference on Image Processing (ICIP)*, pages 538–541, 2005.

[33] K.T. Reed and J.R. Parker. Automatic computer recognition of printed music. *Proceedings of the 13th International Conference on Pattern Recognition (ICPR).*, pages 803–807 vol.3, Aug 1996.

[34] J. V. Mahoney. Automatic analysis of music score images. B.Sc thesis. Technical report, Department of Computer Science and Engineering, MIT, 1982.

[35] D. Prerau. *Computer pattern recognition of standard engraved music notation*. PhD thesis, Massachusetts Institute of Technology, 1970.

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

[37] Anjan Dutta, Umapada Pal, Alicia Fornes, and Josep Llados. An efficient staff removal approach from printed musical documents. In *Proceedings of the 2010 20th International Conference on Pattern Recognition*, ICPR '10, pages 1965–1968, Washington, DC, USA, 2010. IEEE Computer Society.

[38] I. Leplumey, J. Camillerapp, and G. Lorette. A robust detector for music staves. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 902–905, 1993.

[39] Alicia Fornés, Anjan Dutta, Albert Gordo, and Josep Lladós. The international conference on document analysis and recognition(icdar) 2011 music scores competition: Staff removal and writer identification. In *International Conference on Document Analysis and Recognition(ICDAR)*, pages 1511–1515. IEEE, 2011.