Restauro de Documentos Antigos por Processamento de Imagem

Pedro Daniel da Rocha Melo e Castro

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática e de Computadores

Setembro de 2007
Abstract

Degradation in ancient documents has been a matter of concern for a long time. With the easy access to information provided by technologies such as the Internet, new ways exist for consulting ancient documents without exposing them to more dangers of degradation. One of those types of documents is written ancient music. These documents suffer from multiple kinds of degradation, where bleed-through outstands as the most damaging one.

This thesis, carried out in collaboration with the Portuguese National Library, encompasses the subject of image processing with the goal of researching approaches to effectively perform written ancient music restoration. Restoration serves not only the purpose of revamping the visual appearance of ancient documents, but also the intention of improving further image recognition operations. A study was realized in order to determine the degradation problems images of ancient music usually portray. The literature was revised and research conducted, leading to the development of a new method that is based on both existing and new researched techniques. The developed method is flexible, in order to handle the varying conditions found in institutions like libraries, providing a good restoration quality in the majority of the cases. The restoration method was implemented in conformance with the requirements of the Portuguese National Library.

Keywords: Ancient Music Restoration, Image Processing, Document Degradation, Bleed-through Removal, Digital Libraries.
Resumo

A degradação de documentos antigos tem sido matéria de preocupação por muito tempo. Com a facilidade de acesso à informação proporcionada por tecnologias como a Internet, novas formas existem de consultar documentos antigos sem os expor a ainda mais perigos de degradação. Um desses tipos de documentos é a música manuscrita antiga. Estes documentos sofrem de múltiplas formas de degradação, onde o "bleed-through" sobressai como aquela que mais danifica.

Esta tese, levada a cabo em colaboração com a Biblioteca Nacional Portuguesa, envolve o domínio do processamento de imagem com o objectivo de investigar formas de efectivamente realizar o restauro de música manuscrita antiga. O restauro serve não apenas o propósito de aperfeiçoar a aparência visual de documentos antigos, mas também a intenção de melhorar futuras operações de reconhecimento em imagem. Um estudo foi realizado de forma a determinar quais os problemas de degradação que as imagens de música antiga normalmente apresentam. A literatura foi revista e investigação conduzida, levando ao desenvolvimento de um novo método que é baseado não apenas em técnicas existentes mas também em novas técnicas que foram desenvolvidas. O método é flexível, de forma a lidar com diferentes condições encontradas em instituições como bibliotecas, proporcionando uma boa qualidade de restauro na maioria dos casos. O método de restauro foi implementado em conformidade com os requisitos da Biblioteca Nacional Portuguesa.

Acknowledgements

I would like to express my profound gratitude to all those who supported me in the accomplishment of this thesis. To Prof. João Rogério Caldas Pinto, for his dedication, understanding and thoughtful guidance. To Prof. José Luís Brinquete Borbinha, for his support and for creating the possibility to work with the Portuguese National Library in the first place. To the Portuguese National Library and the Biblioteca Geral da Universidade de Coimbra, for providing valuable image resources. To João Gil and Rui Almeida, for their insight in different parts of this work. To all those at IDMEC who were of assistance during our meetings. Last but certainly not least, to my family and friends, for their loving support and for providing the environment that encouraged me to pursue my goals and go one step further, with this thesis.
# Table of Contents

Abstract i

Resumo ii

Acknowledgements iii

List of Figures vi

List of Tables vii

1 Introduction 1

2 Degradation 3

3 Color Image Segmentation 6

  3.1 Feature-Space Based Segmentation ........................................ 7
  3.2 Image-Domain Based Segmentation ........................................ 8
    3.2.1 Area-Based Segmentation .............................................. 9
    3.2.2 Edge-Based Segmentation ............................................. 9
    3.2.3 Area and Edge Based Segmentation Integration ..................... 10
  3.3 Physics-Based Segmentation .............................................. 11
  3.4 Summary ................................................................. 11

4 Background Homogenization 13

  4.1 State of the Art ....................................................... 13
  4.2 Methods ............................................................... 14
  4.3 Experiments ........................................................... 17
  4.4 Summary ............................................................... 21

5 Bleed-through Removal 23

  5.1 State of the Art ....................................................... 23
  5.2 Bleed-through Restoration Method ...................................... 28
  5.3 Feature Extraction and Classification Heuristics ..................... 31
  5.4 Experiments ........................................................... 35
  5.5 Summary ............................................................... 40

6 Software 42

  6.1 Restoration toolbox ............................................... 42
  6.2 The Complete Restoration Method ...................................... 42
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Images showing water blotches.</td>
<td>3</td>
</tr>
<tr>
<td>2.2</td>
<td>Images showing bleed-through besides valid musical symbols.</td>
<td>4</td>
</tr>
<tr>
<td>2.3</td>
<td>Images showing exclusively bleed-through.</td>
<td>4</td>
</tr>
<tr>
<td>2.4</td>
<td>Images showing text annotations and a superimposed stamp.</td>
<td>4</td>
</tr>
<tr>
<td>2.5</td>
<td>Images showing broken musical symbols.</td>
<td>5</td>
</tr>
<tr>
<td>2.6</td>
<td>Images showing global degradation.</td>
<td>5</td>
</tr>
<tr>
<td>4.1</td>
<td>Outline of the Canny edge detection with post-processing method.</td>
<td>16</td>
</tr>
<tr>
<td>4.2</td>
<td>Images resulting from the multiple stages of the Canny edge detection method with post-processing.</td>
<td>17</td>
</tr>
<tr>
<td>4.3</td>
<td>Samples of the set of 10 images used throughout the tests.</td>
<td>18</td>
</tr>
<tr>
<td>4.4</td>
<td>Average precision, recall and $g$-mean results for the tested methods.</td>
<td>19</td>
</tr>
<tr>
<td>4.5</td>
<td>$g$-mean results for the tested methods, in each of the 10 images.</td>
<td>20</td>
</tr>
<tr>
<td>4.6</td>
<td>Result of processing two of the original images with the chosen methods. Columns 1 and 2 correspond to the processing of images 2 and 3, respectively. Presented from top to bottom are: the original images; the result of binarizing the original images with Niblack thresholding, Sauvola thresholding, Otsu thresholding, Canny edge detection with Niblack post-processing, Canny edge detection with Sauvola post-processing, and Fuzzy C-Means; and the restored images using Sauvola thresholding, the method that performed the best.</td>
<td>22</td>
</tr>
<tr>
<td>5.1</td>
<td>Activity diagram for the bleed-through restoration method, including its input and output objects.</td>
<td>29</td>
</tr>
<tr>
<td>5.2</td>
<td>Samples of the set of 14 images used throughout the tests.</td>
<td>36</td>
</tr>
<tr>
<td>5.3</td>
<td>Average precision, recall and $g$-mean results for the tested heuristics.</td>
<td>39</td>
</tr>
<tr>
<td>5.4</td>
<td>$g$-mean results for the tested heuristics, in each of the 7 image pairs.</td>
<td>39</td>
</tr>
<tr>
<td>5.5</td>
<td>Result of processing two of the original images with the chosen heuristics. Columns 1 and 2 correspond to the processing of images 2 and 4, respectively. Presented from top to bottom are the original images and the result of removing bleed-through using the MinDiff, ThreshDiff, CCMinDiff, DuboisDano and MinCorr heuristics, respectively.</td>
<td>41</td>
</tr>
<tr>
<td>6.1</td>
<td>Activity diagram for the complete restoration method, including its input and output objects.</td>
<td>46</td>
</tr>
</tbody>
</table>
List of Tables

3.1 Summary of color image segmentation types. .................................................. 12

4.1 Detailed precision (P), recall (R) and g-mean (G) results obtained by applying 6 different methods to 10 images of ancient music. .................................................. 20

5.1 Detailed precision (P), recall (R) and g-mean (G) results obtained by applying 11 different heuristics to 7 pairs of images of ancient music. .................................................. 37

6.1 Signature and summary of the background homogenization functions contained within the toolbox. .................................................. 43

6.2 Signature and summary of the bleed-through functions contained within the toolbox. ........ 44
1 Introduction

Documents of ancient music frequently appear with multiple signs of degradation. Most of these signs result from the aging process, especially if poor care was taken to store and preserve the documents over time.

It is becoming a common task in institutions like libraries to provide an easy access to different types of information. Making use of technologies such as the Internet, libraries all around the world enable people to have contact with rare and ancient documents that would have been otherwise unavailable. To this intent, the process usually evolves from the creation of high resolution images, using image scanners, to the generation of different versions of these images, differing in aspects like image quality and size. These images are then distributed accordingly.

One common feature to all of the digitized images is, however, that they continue to reveal the deterioration that was present in the original documents. Due to the aging process, documents suffer multiple transformations and may become almost imperceptible. When handling the images of those documents, it is therefore desired to have these images restored in order to revamp their visual appearance. Moreover, restoration is also needed in order to improve further image segmentation and recognition operations.

Objectives

The work in this thesis was carried out in collaboration with the Portuguese National Library, which supplied the necessary resources and served as the final target for the application of the techniques here presented.

The primary objective within this thesis is to provide an original contribution to the field of image processing, in particular to solve the problem of degraded ancient music documents using restoration techniques. To this purpose, foundations are established in the context of image processing and restoration, the literature is reviewed with regard to the applicability of the existing approaches to the context of ancient music, and methods are proposed and analysed in order to successfully solve the problem.

The subsequent objective is to deliver a complete software which contains the best methods hereby presented. These methods are those determined to perform the best under the conducted study and experimentation.

Characteristic of these objectives is naturally the notion of their feasibility. The intent is to provide a means for restoration that works most of the times, with the realistic consciousness that performing restoration that works every time, with every image and on any condition is far from possible.

These are, in summary, the main contributions of this thesis: 1) a state of the art survey of methods related to the restoration of ancient documents, in general, and written music, in particular; 2) the development of a complete ancient music restoration method; and 3) the implementation of the restoration method for use in the Portuguese National Library.

Organization

Presented in Chap. 2 is an overview of degradation types that are observable in images of ancient documents. Chapter 3 provides a basis for color image segmentation, including a categorized description and comparison of existing segmentation approaches.

Chapters 4 and 5 detail the fundamental studies realized in this thesis. Chapter 4 discusses general methods for ancient music restoration, while Chap. 5 discusses methods for bleed-through removal. Each of these chapters provides: 1) a state of the art survey of existing methods; 2) a discussion on the applicability of these methods in the context of ancient music; 3) a selection of both existing and new proposed methods to be used throughout experiments; 4) the description of the experiments; and 5) the analysis of the results.
Chapter 6 provides a description of the software developed throughout this work. This includes the software developed for experiment purpose as well as the final software to be used in the Portuguese National Library. Lastly, Chap. 7 concludes this dissertation, indicating its main contributions and future work directions.
2 Degradation

Degradation can be seen as “every sort of less-than-ideal properties of real document images, e.g. coarsening due to low digitizing resolution, ink/toner drop-outs and smears, thinning and thickening, geometric deformations, etc” [Bai00]. Restoration, on the other hand, can be thought as a transformation process that gives the original aspect to images that show a certain state of degradation. Restoration is needed not only to improve the appearance of a document but also to improve the results of further segmentation and recognition operations. By clearing artifacts from the images there is less room, in the future, for misinterpretation.

A document can appear degraded in multiple ways. The reasons for the degradation may vary from poor source type and the image acquisition process to the storage environment that directly causes problems for the image quality. Degradation is unquestionably one of the main reasons for image processing to fail. Most degradation types in document images affect both physical and semantic understandability in the document analysis tasks, such as page segmentation, classification and optical character recognition.

A typology for different types of degradation on old document images has been proposed by Drira [Dri06]. This typology was preceded by an exhaustive research of all degradations, which was done by consulting various images of degraded documents. The proposed classification was made according to the further treatment that will be applied in the context of virtual document image restoration. It is decomposed into three classes: 1) background degradation; 2) foreground degradation; and 3) global degradation. These will be described next, along with examples of typical degradations that appear on ancient music documents. All images are courtesy of the Biblioteca Universitária de Coimbra, Portugal.

Background Degradation

The most common degradation is characterized by the presence of artifacts in the background of documents. It includes blotches due to humidity, marks resulting from ink that traverses the paper (bleed-through) or from the scanning process (show-through), underlines, strokes of pen, annotations, and the superimposition of other symbols.

Ancient music documents generally comprise a combination of these degradations. Water blotches (Fig. 2.1) are characterized by having a mainly convex shape (due to the diffusion of water molecules in the paper), a color that is darker than the neighborhood (due to the dust which is attracted in the paper texture), and an even darker border area where the dust accumulates [SR05]. Bleed-through (Fig. 2.2) refers to the sipping of ink from one side of a page to the other. It can be quite damaging, showing intensity levels that can be even darker than the true valid musical symbols in the foreground. Fig. 2.3 presents images containing exclusively bleed-through, to demonstrate how damaging it can be. Underlines are more frequent in text documents than they are in music documents, but strokes, annotations and superimposed symbols do appear quite often (Fig. 2.4).

Foreground Degradation

Degradation on the foreground generally leads to broken or touching foreground objects, for instance, characters or musical symbols. Age effects can affect the ink components of a document. Many chemical effects can occur leading to ink disappearance and some gaps can even appear in the document image causing

![Figure 2.1: Images showing water blotches.](image)
significant loss of data and therefore affecting the document's content. For instance, gaps create regions with homogeneous colors. They incur a complete loss of data, whereas semitransparent blotches can preserve part of the original data. In the case of text document treatment, filling in the gap is rather complicated to proceed. Sometimes, historians must contribute. Foreground degradations are also sometimes introduced when attempting to correct musical symbols, either by covering them with white paint or by scratching them. Degradations of this kind are shown in Fig. 2.5.

**Global Degradation**

This type of degradation affects documents in their entirety. It refers to a transformation that can be observed in a document as a whole, i.e., without affecting uniquely the foreground or the background. This transform can act either on the localization of the pixel (skew, degraded curve) or on its value (transformation of the color). Geometrical degradation is a type of global degradation very common when scanning thick documents. The paper surface of these documents in the course of scanning is usually curved. The presence of this curvature leads to warped words appearing around the book spine area and to a non-uniform illumination. Time effect is another degradation included in this class. Fig. 2.6 presents samples of global degradation in ancient music.
Figure 2.5: Images showing broken musical symbols.

Figure 2.6: Images showing global degradation.
3 Color Image Segmentation

Image segmentation is a fundamental operation in Computer Vision [HS92] which consists on subdividing an image into its constituent parts, under a certain criteria. Machine vision systems are often considered to be composed of two subsystems: low-level vision and high-level vision. Low-level vision consists primarily of image processing operations performed on the input image to produce another image with more favorable characteristics. High-level vision includes object recognition and, at the highest level, scene interpretation. The segmentation system can be seen as the bridge between these two subsystems. Through segmentation, the enhanced input image is mapped into a description involving regions with common features which can be used by the higher level vision tasks [Spi93].

Image segmentation should produce results that clearly distinguish different areas of interest within an image, while being the least affected by noisy and irrelevant objects. These different areas are acknowledgeable as conforming to a certain criteria. Haralick and Shapiro [HS85] state the ideal characteristics of a good segmentation: “Regions of an image segmentation should be uniform and homogeneous with respect to some characteristics such as gray tone or texture. Region interiors should be simple and without many small holes. Adjacent regions of a segmentation should have significantly different values with respect to the characteristic on which they are uniform. Boundaries of each segment should be simple, not ragged, and must be spatially accurate”.

In a formal manner, image segmentation can be represented as follows [Spi93, LM01, FM81, PP93, JKS95]:

If \( I \) is the set of all pixels and \( P() \) is a uniformity predicate defined on groups of connected pixels, a segmentation of \( I \) is a partitioning set of connected subsets or image regions \( R_1, R_2, \ldots, R_n \) such that

\[
\bigcup_{i=1}^{n} R_i = I, \text{ where } R_i \cap R_j = \emptyset, \quad \forall i \neq j,
\]

(3.1)

and the uniformity predicate satisfies

\[
P(R_i) = True, \quad \forall i,
\]

(3.2)

\[
P(R_i \cup R_j) = False, \quad \forall R_i \text{ adjacent to } R_j,
\]

(3.3)

\[
(R_i \supset R_j) \land (R_j \neq \emptyset) \land (P(R_i) = True) \Rightarrow P(R_j) = True.
\]

(3.4)

Several surveys exist that focus on monochrome image segmentation [Spi93, HS85, FM81, PP93, SSW88, Che81]. Cheng et al. [CJSW01] describe the major image segmentation techniques covered in each of these surveys. In what concerns monochrome segmentation, the approaches are based on the gray levels of each pixel that form an image. In particular, these approaches are based on their homogeneity, discontinuity or in a combination of both. The former includes clustering, thresholding, region growing, and region splitting and merging. Approaches based on discontinuity identify regions by detecting abrupt changes in intensities, and include edge-based methods. A third approach consists in improving the results of the homogeneous and discontinuity-based approaches by combining them [PL90].

Quoting Henri Matisse: “Before, when I didn’t know what colour to put down, I put down black. Black is a force: I depend on black to simplify the construction. Now I’ve given up blacks”. Image segmentation started with grayscale images. It was simpler to start with, but mostly the technology at the time did not allow for the added complexity of processing color components. As technology evolved, the possibility and necessity of dealing with the extra characteristics that colors provide became evident. Most monochrome image segmentation techniques were extended to color images. Each color component of an image can be interpreted as an intensity image and used in a gray level segmentation. After applying the segmentation, the components
can be merged back together to form a color image [YT96]. However, questions arise as to which color representation to choose and how to employ color information as a whole for each pixel [CJSW01].

Color image segmentation is defined by Skarbek [SK94] as “a process of extracting from the image domain one or more connected regions satisfying uniformity (homogeneity) criterion which is based on feature(s) derived from spectral components. These components are defined in a chosen colour space model. The segmentation process could be augmented by some additional knowledge about the objects in the scene such as geometric and optical properties”.

Important to image segmentation is the definition of region, for which many variants can be found:

1. Region is a connected component of a pixel set specified by a class membership function defined in a color space.
2. Region is a (maximal) connected set of pixels for which a uniformity condition is satisfied.
3. Region is a connected set of pixels bounded by edge pixels creating a color contour.
4. Region corresponds to a surface or an object of homogeneous material.

The first category is usually referred to as pixel or feature-space based segmentation. The second is typically entitled area or region based segmentation. The third is generally denominated edge or boundary based segmentation. The last definition corresponds to physics-based segmentation and is the only not to inherit from the field of gray-level image segmentation. The second and third region definitions are also sometimes grouped together in a so-called image domain based segmentation.

Many surveys exist that target on color image segmentation [LM01, PP93, CJSW01, SK94, CJS98, LM99, FMR’02]. Most refer to edge and area based segmentation. Some also refer to feature-space based segmentation, while only a few consider physics-based segmentation techniques. Here, the most important aspects of each segmentation type are described. They are categorized in: 1) feature-space based techniques; 2) image-domain based techniques; and 3) physics-based techniques. Furthermore, image-domain based techniques are subdivided into: 1) area-based techniques; 2) edge-based techniques; and 3) the combination of the previous two techniques.

It should be noted that the objective in this chapter is to divide color image segmentation methods into categories, and to describe each of those categories. This therefore entails the description of types of methods, and not the description of the methods themselves.

### 3.1 Feature-Space Based Segmentation

Feature-space based segmentation was well introduced by Lucchese [LM01]:

If we assume that color is a constant property of the surface of each object within an image and we map each pixel of the color image into a certain color space, it is very likely that different objects present in the image will manifest themselves as clusters or clouds of points. The spreading of these points within each cluster is mainly determined by color variations due to shading effects and to the noise of the acquisition device. On the other hand, if instead of mapping pixels into color spaces, we build some ad hoc histograms upon color features, such as hue, for instance, it is likely that the objects will appear as peaks within these histograms.

This segmentation type groups pixels according to a certain feature space, which can be one of the available color spaces or a different color space induced by other color attributes. Moreover, and this is especially important, segmentation techniques of this kind generally discard spatial information. This means that two pixels may share the same group within a feature space (e.g., they may share the same color) even if they are
distant from each other within an image. The usefulness of this approach becomes relevant with the need to distinguish parts of an image with somewhat clear color differences.

The main feature-space based segmentation techniques described here are divided into: 1) histogram thresholding; and 2) color space clustering.

**Histogram Thresholding**

This technique has been used for a long time in gray level image segmentation. In this segmentation type, the intensity levels of an image are used to form an histogram. This histogram will contain peaks for the pixel intensities that are more frequent in the image. The valleys between those peaks can be used as thresholds to divide the image into a number of intensity ranges.

There are several papers that compare and evaluate thresholding techniques [SSW88, TJ95, Alb93, LCP90, Ots79, PSS86]. A common problem with these techniques is that noise often interferes with the segmentation, giving rise to spurious peaks and thus creating ambiguities. Smoothing techniques are usually adopted to minimize this effect [LM01].

Important in the thresholding context is the selection of adequate threshold values. According to the way these values are computed, thresholding techniques can be:

- **Global** – threshold values are calculated for the entire image.
- **Local** – threshold values are calculated for each pixel, based on the information contained within their neighborhood.
- **Hybrid** – threshold values are the result of combining global and local techniques.

When referring to color images, there is the additional complexity of having more than a single component of intensity levels describing an image, since they all have to be combined in some way. Some algorithms independently threshold each color component and combine the results afterwards [CdH98]. Watershed algorithms are also used to partition histograms [SPK98, DL77]. Other approaches include the use of a single color component (e.g., hue or intensity) in the segmentation process [TLT95, TC94, APV98].

**Color Space Clustering**

Quoting Fu and Mui [FM81], “clustering of characteristic features applied to image segmentation is the multi-dimensional extension of the concept of thresholding”.

Clustering in color space collects pixels into groups according to their similarity. Groups, or clusters, are created based on the principle of maximizing the intraclass similarity and minimizing the interclass similarity. Objects within a cluster should have high similarity between each other, but should also be very dissimilar to objects in other clusters [HK00].

A difficulty that arises in the use of clustering for color segmentation is the selection of the ideal number of clusters in an unsupervised clustering scheme, also known as cluster validity. Moreover, the chosen color space can be of great influence to the results. For instance, when using the RGB color space, a color image with shadows or shading may not be segmented properly [CJSW01], with these objects being segmented as separate from those they are the shadow or shade of.

### 3.2 Image-Domain Based Segmentation

Segmenting in the image domain refers to partitioning an image by taking spatial information into account, besides the actual color features. If only color features are used, as described for feature-space segmentation,
results will be very similar in color but possibly spatially disconnected. On the other hand, if only spatial information is applied, results will much likely contain regions that are spatially well connected but heterogeneous in a feature space. Techniques based on the image domain exploit the surface coherence of images [JKS95], i.e., the fact that an object usually has its points spatially close. Techniques based on the image domain can be categorized into area or edge based. Both categories are described next, along with approaches that are based on their integration.

3.2.1 Area-Based Segmentation

This approach to segmentation creates areas of pixels by joining those that are spatially close and similar in features. Available techniques can be further subdivided into two groups: 1) region growing; and 2) split and merge.

Region Growing

As its name implies, region growing grows regions starting with a set of uniform regions called seeds. A similarity criteria is chosen based on the specific problem under consideration. This criteria is applied to each of the seeds in a process of selectively appending to each seed its neighboring pixels that have properties similar to it. The growth stops when no more points can be added to the regions. The stopping rule in use can greatly improve the power of a region growing algorithm. Additional criteria utilizes the concept of size, likeness between a candidate and the pixels grown so far, and the shape of the region being grown [GW02].

The result of region growing techniques highly depend on the choice of seeds. If seeds are not instantiated in the regions of interest, some regions may never be included in the segmentation result. Therefore, these techniques are mainly used for processing single regions. Moreover, the segmentation result is also influenced by the order in which pixels and regions are examined. After a region growing process, there may exist some very small regions or there can be two or more neighboring regions grown at different times exhibiting similar attributes. A common post-processing provision therefore consists in a merging phase that eliminates such instances by generating broader regions [LM01].

Split And Merge

With split and merge, an entire image is used as the initial seed. This initial seed is most probably a nonuniform region, and as such it is split into a number of subregions, typically four, equal-sized and squared. For any homogeneous subregions found, merging heuristics are applied to join them together with other homogeneous adjacent subregions, in such a way that maximizes the uniform area. The process is repeated for all remaining nonuniform regions. The homogeneity criteria is generally based on the analysis of the chromatic characteristics of the region. A region with small standard deviation in the color of its members is considered homogeneous.

3.2.2 Edge-Based Segmentation

Contrasting with area-based segmentation, which forms regions by detecting continuities of pixels on the image, edge-based segmentation divides an image into regions based on discontinuities. While in gray-level segmentation these discontinuities always correspond to changes in the intensity levels, in a color image an edge is identified as a discontinuity in a three-dimensional color space. In a matter of fact, there can be major color changes in the pixels without those changes being reflected in their corresponding gray levels, so these changes can only be detected using segmentation in the color space. Novak and Shafer [NS87] state that
gray-scale edge detection can account for about 90% of the total edge points in a color image. Color edge detection is therefore required for the remaining 10%.

Color edge detectors constitute a critical part of color edge segmentation. They can be divided into techniques that embed in a single measure the variations of all three color channels, and techniques that compute the gradients of the single channels and combine them according to a certain criteria. The later can be further categorized into: 1) output fusion methods; 2) multidimensional gradient methods; and 3) vector methods [RT01].

Besides detecting the location of edge points, it is very important to be able to link these points to define actual region limits. Some authors distinguish the concept of edge from the concept of boundary [GW02]. An edge is considered a local concept, while a boundary is described as a global idea. Local techniques determine the location of edge points by performing operations based solely on the neighborhood of those points. This means that the detection performed on a specific point is completely independent from the detection performed in the rest of the points of an image. When linking edge points, those that are similar according to a certain criteria are linked together, forming an edge based on that criteria. Global techniques make a sort of global optimization, and therefore a certain edge point can be identified after many optimization steps involving changes in large areas. When it comes to processing the edge points, they are linked by first determining if they lie on a curve with a certain shape. This approach evidently takes into account the global relationship between pixels.

3.2.3 Area and Edge Based Segmentation Integration

The two main methods previously described for image-domain segmentation often fail to produce accurate segmentation results [PL90]. With the aim of improving the segmentation process, a large number of algorithms which integrate both area and edge-based segmentation have been proposed. One of the main differences between these algorithms is the time at which integration is performed: 1) embedded in the area detection; or 2) after both area and edge segmentation [FBC94].

Embedded Integration

This type of integration can be used as defining new parameters or a new decision criteria for segmentation. In the most usual approach, edge-based segmentation is first performed, and its results are then used in area-based segmentation.

It was previously referred that area-based segmentation highly depends on the way the initial regions are formed and also on their growing criteria. Therefore, there are two main tendencies in which edge information is used to improve segmentation based on areas [FMR*02]:

1. Control of decision criterion – edge information is used in the decision criteria that controls the way regions grow.
2. Guidance of seed placement – edge information is used to assist on the placement of the initial seeds.

Post-Processing Integration

This type of integration is performed a posteriori to the independent segmentation of an image using the different area and edge based approaches. Area and edge information is first extracted and then integrated. Post-processing integration is based on fusing results from single segmentation methods attempting to combine the map of regions (generally with thick and inaccurate boundaries) and the map of edge outputs (generally with fine and sharp lines, but dislocated), with the aim of providing an accurate and meaningful segmentation [FMR*02].

Three different post-processing approaches can be identified:
1. Over-segmentation – a segmentation method with specifically fixed parameters is used to obtain an over-segmented result. Additional information from other segmentation techniques is used to eliminate false boundaries which do not correspond to regions.

2. Boundary refinement – considering area-based segmentation as a first approach, with well defined regions but inaccurate boundaries, this approach uses edge information to refine the obtained region boundaries and create a more precise result.

3. Selection-evaluation – edge information is used to evaluate the quality of different region-based segmentation results, with the intent of choosing the most reliable one. It deals with the difficulty of establishing adequate stopping criteria and thresholds for segmentation.

### 3.3 Physics-Based Segmentation

Segmentation techniques of this kind belong to a new class of computer vision methods that use reflection models based on the properties of the material in the scene.

All the previous segmentation approaches are ineffective when applied to images with light variations such as highlights, shadowing and shadows. These occurrences cause objects with uniform colors to appear with color variations, which can result in over-segmented regions [HSW92]. To overcome this drawback, it is necessary to analyze how light interacts with colored materials and to introduce models of this physical interaction into segmentation algorithms.

Colored materials can be divided into three main categories: 1) optically inhomogeneous dielectrics; 2) optically homogeneous dielectrics; and 3) metals. The former includes materials such as paints, papers, plastics and textiles. The second includes materials like glass and crystals. The third is considered self-explanatory.

The mathematical methods used in physics-based segmentation do not vary significantly from those already described in the previous sections. They differ on the underlying physical models which now account for the reflection properties of the materials. The most common physics models are the dichromatic reflection model [Sha85] and the unichromatic reflection model [Hea89]. The former has become very popular, mostly due to its simplicity and effectiveness of representation.

The existing physics-based models are only efficient when processing materials whose reflection properties are known and easy to model. They have very rigid assumptions in what concerns the material type, the light source and illumination [CJSW01]. Therefore, their reliability only allows them to be used in a limited set of applications.

### 3.4 Summary

A categorized description of color image segmentation types, as existent in the literature, has been presented and is summarized in Table 3.1. Not only do different types of segmentation appear with different purposes, they also relate to the context of ancient music in different ways. This will be visible in the next chapters.
<table>
<thead>
<tr>
<th>Type</th>
<th>Summary</th>
<th>Subtype</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature-based</td>
<td>Groups pixels according to a feature space (e.g., color values). Generally discards spatial information.</td>
<td>Histogram Thresholding</td>
<td>A histogram is formed from the intensity levels and divided into parts by the means of thresholds. Thresholding can be global (one threshold for the entire image), local (one threshold per pixel) or hybrid.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Color Space Clustering</td>
<td>Collects pixels into clusters according to their similarity. Clusters are created based on the principle of maximizing the intraclass similarity and minimizing the interclass similarity.</td>
</tr>
</tbody>
</table>
4 Background Homogenization

This chapter discusses general approaches to the restoration of ancient music, where general means that even though they are customized to the specific context of ancient music documents, they are not specific to a single kind of deterioration. As will be observed, while these approaches provide good results in the general cleaning of music documents, they are unsatisfactory in what corresponds to the removal of bleed-through. This kind of extreme deterioration therefore requires a specialized treatment, which is the concern of Chap. 5.

Homogenizing the background consists of detecting which parts of an image correspond to the true foreground. This includes distinguishing the foreground from all background and interference. Interference in the background of a document can be generally detected using segmentation in the feature space, as described in Sect. 3.1. Because the background interference usually presents intensity levels that are a mid-term between the background and the foreground, thresholding and clustering methods can be used to separate these defects from the rest of the image. However, there are also cases of extreme degradation in which more sophisticated methods have to be used.

This chapter is organized as follows. First, the existing methods for background homogenization are surveyed. A discussion is then performed in order to determine which of the existing methods are plausible for the restoration of ancient music. These methods are described and become the subject of experimentation, for which the methodology is presented and the results detailed and examined.

4.1 State of the Art

Presented here are known methods that relate to the problem of background homogenization. The majority of these methods are targeted on images of text documents, while others exist for images of maps and artworks.

Many segmentation approaches that aim to extract clear text from either noisy or textured backgrounds have been reported and compared in the literature [CL96, NKHW99, LS97, LAS94, WR83, GPP04, SP00, Don01, LVP02, GPH06]. Negishi et al [NKHW99] described an automatic thresholding algorithm to extract character bodies from noisy backgrounds. Their algorithm removes black frames that surround page areas and equalizes the background to eliminate the overlapping between the gray levels of character parts and those of the background. An automatic thresholding method is then applied to convert the original image into a reasonably noise-free binary image. Liu and Srihari [LS97] used a thresholding algorithm based on texture features to extract characters from run-length featured texture backgrounds. Their algorithm utilizes two fundamental attributes of document images: 1) the characters normally occupy a separable gray-level range in the grayscale histogram; and 2) the text images contain highly structured stroke units. Liang et al [LAS94] presented a morphological approach to extract text strings from regular periodic overlapping text-background images, useful for removing backgrounds with repetitive patterns. Don [Don01] used noise attribute features based on a simple noise model to overcome the difficulty that some objects do not form prominent peaks in the histogram encountered by many conventional global thresholding methods. Leedham et al [LVP02] compared several thresholding techniques for separating text and background in degraded historical documents.

Clustering has also been used to detect interference. Pinto et al [PBSP05] proposed a method that combines fuzzy clustering with mathematical morphology in order to extract interference from images of old text documents. Using a Fuzzy C-Means algorithm with 3 clusters, an image is partitioned into background, foreground and interference. Mathematical morphology is afterwards used to remove underlines and small artifacts that may still be present within the detected foreground.

Edge-based segmentation has also been reported to work. Tan et al [TCWS00, TCS02] consider the extraction of a clean foreground from historical handwritten documents. Their method is specific to bleed-through
removal, so it is thoroughly described in Sect. 5.1, however part of it can be used for background homoge-
nization, hence being included here. With the observation that edges of the valid foreground are sharper than
those of the interference, edge detection algorithms can be used to detect the foreground edges and discard
the degradation.

Other approaches exist that attempt to recover from defects found in artworks [SR05, BBC00, dRBCB01,
SRT03, SRdP03]. In particular, these approaches have been developed for the removal of water blotches and
cracks found in paintings and photographic prints. They conclude that it is very difficult, if not impossible, to
automatically detect these defects as there are no simple rules to distinguish them from real image features.
As such, they follow a semi-automatic approach. A user is required to interactively identify and select the
defective areas of a picture, after which algorithms are used to detect the boundaries of the defects, starting
with the points selected by the user.

Applicability

The problem of ancient music restoration finds itself with many constraints that inhibit the application of most
of the described methods. The inherent characteristics of the existing methods target them on the solution of
a limited set of problems, most of which do not adapt to the problem of ancient music. As a consequence,
only a subset of these methods can actually be used in the context of ancient music restoration.

To deal with the problem of heterogeneous backgrounds, segmentation in the feature space is a reasonable
approach. Because the background interference usually presents intensity levels that are between those
the background and the foreground, thresholding and clustering methods can be used to separate these
defects from the rest of the image. Methods based on extracting clear text from either noisy or textured
backgrounds do not succeed when applied to ancient music. The reason is twofold: 1) most use text features
to identify the valid foreground parts; and 2) some use textures or patterns to remove noisy background, but
degraded background regions cannot be modeled this way as they constitute not really a noise pattern but
rather distinctive regions.

Cases of extreme background degradation, however, require different approaches to be taken into account.
It can be observed that most of the background interference appears in the form of an area with smooth
contours. As such, the use of edge detection may be able to detect it, as the valid foreground typically
presents sharper contours. This method can also be combined with the previous ones, i.e., using thresholding
or clustering after edge detection.

4.2 Methods

The methods selected to be used throughout the experiments are described next.

Niblack Thresholding

Niblack's thresholding method [Nib86] performs adaptive thresholding and was selected because it is fre-
quently cited and has been thoroughly reviewed with other types of documents. Comparisons have revealed
that it outperforms other thresholding algorithms under different application domains [TJ95, TT95, HDDK05].

This algorithm calculates a threshold value for each pixel based on the mean and standard deviation of all the
pixels in a local neighborhood of it. A window of size $N \times N$ is moved over the image and the threshold value,
for a pixel $(x, y)$, is calculated as

$$ t(x, y) = m(x, y) + K \cdot s(x, y) $$

(4.1)

where $m(x, y)$ and $s(x, y)$ are the mean and standard deviation values, respectively, in a local neighborhood of
size $N \times N$ of pixel $(x, y)$. 
The parameters $N$ and $K$ highly influence the segmentation result. The window size $N$ must be small enough to keep the details, but also large enough to remove noise. The value of $K$ is used to decide how much of the total print object boundary is retained. A difficulty naturally arises on how to determine the best $N$ and $K$ parameters. The parameters $N$ and $K$ were tested ranging from 9 to 71 and from −0.2 to −1, respectively. The best parameter set did vary from image to image, but the values of $N = 51$ and $K = −0.8$ were found to average better.

**Sauvola Thresholding**

Sauvola’s thresholding method [SP00] performs adaptive thresholding and was chosen because it is a modification of Niblack’s method aimed at dealing better with the cases in which the background contains light texture, big intensity variations and uneven illumination. These properties are characteristic of images of ancient music.

In this modification, the threshold values are computed with the dynamic range of standard deviation, as

$$t(x, y) = m(x, y) \cdot \left[ 1 + K \cdot \left( \frac{s(x, y)}{R} - 1 \right) \right].$$

(4.2)

where $m(x, y)$ and $s(x, y)$ are still the mean and standard deviation values, respectively, in a local neighborhood of size $N \times N$, as they were in Niblack’s formula. The parameter $R$ is the dynamic range of standard deviation and $K$ now assumes positive values. Our tests revealed that the value of $N$ was of limited influence to the quality of the segmentation, except with images containing bigger music note heads, in which case lower $N$ values would not detect their centers. The value of $R$ was fixed to 128, as used by Sauvola [SP00]. From tests performed with $N$ and $K$ ranging from 5 to 31 and from 0.2 to 0.8, respectively, the values of $N = 15$ and $K = 0.2$ provided the best results.

**Otsu Thresholding**

Otsu’s method [Ots79] was included here for completeness, to perform global thresholding. It has been thoroughly evaluated [TJ95, TT95] and used before.

This method selects a threshold that maximizes between-class variance after creating the histogram of the intensity image. This threshold is then applied to all of the image pixels. It has the advantage of not requiring the input of parameters, but assumes that histograms are bimodal and illumination is uniform.

**Fuzzy C-Means Clustering**

This method shares similarities with that of Pinto et al [PBSP05], which uses fuzzy clustering in the restoration of old text documents. Clustering is used in the RGB color space to collect pixels into groups according to their similarity. Groups, or clusters, are created based on the principle of maximizing the intraclass similarity and minimizing the interclass similarity. Objects within a cluster should have high similarity between each other, but should also be very dissimilar to objects in other clusters. Fuzzy clustering distinguishes itself from hard clustering by not assigning an object into a single cluster. Each object can belong to more than one cluster, with specific membership levels.

The Fuzzy C-Means [Bez81] is well known and revealed to present good results in other domains [PBSP05]. It is used to partition an image into $N$ clusters, where each pixel has a membership degree to each cluster. The cluster centers are used to determine which cluster relates to the darkest color. All pixels that share a degree of membership to that darkest cluster that is greater than a value $M$ are selected as valid foreground pixels.
A difficulty arises in the selection of the ideal number of clusters, as well as on the minimum required membership value $M$. From tests with $N$ and $M$ ranging from 2 to 5 and from 0.65 to 0.95, the best results were achieved with $N = 3$ and $M = 0.75$.

**Canny Edge Detection with Niblack Post-processing**

This method is a composition of algorithms, similar in part to what was used by Tan et al [TCWS00], with some variations. As outlined in Fig. 4.1, a Canny edge detector [Can86] is used initially, followed by foreground recovery and an adaptive thresholding algorithm, in this case Niblack’s. Fig. 4.2 depicts the image resulting from the multiple stages of this method, which are described next.

Canny edge detection is used to detect the edges within the image based on the observation that the edges of valid foreground are usually sharper than those of the background interference. This edge detector was selected due to the mentioned previous work with images of text documents, and also because it is known to many as the optimal edge detector, following the assumptions: 1) low error rate; 2) good localization on edge marking; and 3) minimal number of responses for a single edge. This algorithm starts by applying a Gaussian filter to smooth the image. Next, the gradient magnitude and orientation are computed using finite-difference approximations for the partial derivatives. Non-maximal suppression is applied to the gradient magnitude and double thresholding is used to detect and link the edges. The two thresholds are used to eliminate weak edges. Initial seeds relating to strong edges are found using the high threshold. These seeds are grown to the pixels connected to them showing a gradient value above the low threshold. This inherently imposes the difficulty of selecting good threshold values. An approach to threshold selection that is often used is to select the thresholds automatically based on the histogram of gradient magnitudes. The high threshold is set to the level that allows separating 30% of the highest gradient magnitudes from the rest. The low threshold is set to 40% of the high threshold, according to Canny’s recommendations. Fig. 4.2(b) shows an example of detected edges.

Following the detection of the edges, recovery is needed to restore the original foreground. The dilation mathematical morphology operator is used with a structuring element that corresponds to a square of size $D \times D$. Its use has the purpose of filling the regions around the edges. The neighboring pixels within an $D \times D$ window centered on each edge pixel are recovered. The pixels outside the recovered region are set to the average color of the region they fill. Fig. 4.2(c) shows the result of applying the dilation operator to the edges that were previously detected. This image is used to recover the original pixels, resulting in the image shown in Fig. 4.2(d).

As can be observed in Fig. 4.2(d), the dilation operator retains not only valid parts of the foreground but also background and degradation surrounding the edges. Therefore, adaptive thresholding is used to remove the remaining interference. Niblack’s method was once again selected for this task, due to the reasons previously mentioned while describing this adaptive thresholding method. Fig. 4.2(e) presents the result of the binarization process. This binary image can be used together with the original image to retain the original color of the foreground, as shown in Fig. 4.2(f). The background and interference pixels were set to their average color.

The parameters $D$, $N$ and $K$ were tested ranging from 5 to 9, from 9 to 71, and from $-0.2$ to $-1$, respectively, being $N$ and $K$ the parameters for Niblack’s thresholding. In what concerns the $D$ parameter used for the

![Figure 4.1: Outline of the Canny edge detection with post-processing method.](image-url)
dilation, the value of 9 has to be used for the foreground objects to be completely filled. Contrary to what happens with text documents, where a value of 7 has been used [TCWS00], musical objects can many times be thicker thus requiring a higher $D$ value. The values $N = 51$ and $K = -0.8$ provided the best results for the thresholding phase.

**Canny Edge Detection with Sauvola Post-processing**

This method is similar to the previous one, with the difference that Sauvola thresholding is used in the post-processing phase, instead of Niblack’s. From tests with $D$, $N$ and $K$ ranging from 7 to 9, from 5 to 31, and from 0.2 to 0.8, the values of $D = 9$, $N = 15$ and $K = 0.2$ provided the best results, where $N$ and $K$ are the parameters for Sauvola’s thresholding.

### 4.3 Experiments

Experiments were performed in order to determine which method achieves a better segmentation. The emphasis was therefore put on the quality of the restoration as observed in the resultant images.

**Methodology**

A total of 10 images were used throughout the experiments. Samples of these images are presented in Fig. 4.3. The images were selected according to properties considered to be of interest for testing the different methods. These properties include a variety of degradations, as well as different musical notations and illumination characteristics. The images, at different resolutions, were resized to the resolution of 150 dpi. The reason was to ease and quicken the experimentation task, as it involved the generation of thousands of images when testing the methods with its numerous parameter possibilities. All images were first manually restored, using graphics editing software, in order to allow them to be used later as a standpoint for comparison.

The methods used throughout the experiments were developed in the MATLAB environment and programming language. Details on the produced software is presented separately in Chap. 6.

Except for the case of Fuzzy C-Means, which clustered the images in the RGB color space, all other methods had the original images converted to gray scale before processing. This was performed using the known weighted conversion

$$Gray = 0.3 \cdot R + 0.59 \cdot G + 0.11 \cdot B,$$

where $Gray$, $R$, $G$ and $B$ represent the gray, red, green and blue intensity levels, respectively. The result of all methods is a binary image containing the restored foreground and background. After binarization, this image can be applied to the original images as a mask, in order to retain the original colors of the foreground. In what concerns the background, it can be set to a fixed color or the average of its pixels, just to name a few simple inpainting [BSCB00] alternatives.

The test images were processed by the chosen methods and compared to the manually restored images. The comparison was evaluated by the standard measures of *precision* and *recall* [JDH99], with a slight modification. To evaluate text segmentation, these measures are typically used with the precision of a character
or word. When relating to ancient music, this does not apply so well as the musical notation is very varied, including notes, clefs, key and time signatures, rests, bar and staff lines, as well as text, among other symbols. All of these symbols could be treated as characters, for instance, but that would ignore the great differences they have in shapes and sizes. Therefore, a bitwise comparison was performed and, as such, the precision \( P \) and recall \( R \) measures were used as

\[
P = \frac{\text{Correctly Detected Pixels}}{\text{Total Detected Pixels}}, \tag{4.4}
\]

\[
R = \frac{\text{Correctly Detected Pixels}}{\text{Total Pixels}}, \tag{4.5}
\]

where “Correctly Detected Pixels” refers to the foreground pixels that were correctly binarized (i.e., that are equal to those of the manually restored images) by a specific method, “Total Detected Pixels” refers to the total foreground pixels that were binarized by that method, and “Total Pixels” refers to the total foreground pixels that are present in the manually restored image.

Precision and recall reflect the performance of removing interfering strokes and restoring foreground strokes, respectively. To relate the two measures, the geometric mean (g-mean) was used, being defined as

\[
g\text{-mean} = \sqrt{P \times R}. \tag{4.6}
\]

This measure was used because it does not depend on the distribution of examples between classes \([KHM98]\), which is convenient as the number of foreground pixels is typically a minority when compared to the entire set of pixels within the image. This is the actual value that is worth maximizing.
Results

The results of evaluating the selected methods with the set of selected images are presented in Table 4.1. Detailed are the precision ($P$), recall ($R$) and $g$-mean ($G$) values, as well as their averages.

The obtained average values can be easily visualized in the graph of Fig. 4.4. Sauvola’s thresholding was the method to obtain the highest average $g$-mean score, but only slightly better than Canny Edge Detection with Sauvola thresholding post-processing. Niblack’s thresholding was better when used as a post-processor for Canny Edge Detection, but only with a small difference compared to when it was used alone. The Fuzzy C-Means method obtained the highest precision value, but also the lowest recall. This means that most of the pixels it detected as foreground really were foreground, but it only detected a small subset of the total set of foreground pixels, hence having a low recall value. Besides this method, Otsu’s thresholding also presented a great difference between precision and recall averages. In this case, recall was higher than precision, being somewhat near the recall averages of Niblacks’ methods, while precision assumed the lowest average in the tests.

The $g$-mean scores for the 10 selected images is represented in the graph of Fig. 4.5. This allows to describe the regularity of the methods from image to image. It should be noted that even though a line graph was used, the data is discrete and, as such, the lines connecting the data measured for each image only appear to ease the observation of which points belong to the same method, besides the visualization of the regularity in the measured values. As can be observed, Sauvola’s thresholding method was always above the remaining methods, with Canny Sauvola following close. The Canny Niblack and Otsu methods presented the greatest value variations, while Sauvola and Canny Sauvola were the most regular ones. This relates to a degree of predictability which is desired when using these methods. Fuzzy C-Means was more relevant with images in which the degradation showed a strong color. When converted to gray scale, as performed with the other methods, the degradation color may become a gray level in some parts similar to that of the foreground. In this case, clustering in the color space may be able to group the degradation pixels together, distinguishing them from the remaining pixels.

Sauvola’s method was able to outperform Niblacks’ with the much smaller window of size $15 \times 15$, compared to the window of size $51 \times 51$ used with Niblack’s method. In fact, using a $15 \times 15$ window with Niblack’s method resulted in a particularly bad segmentation, with most of the symbols being barely recognizable as most of the interference would be detected as foreground too.

Two of the images used within the experiments are presented in Fig. 4.6, along with the results of processing them with the chosen methods. Image 2 presents less varying results, but still noticeable interference is ob-

![Figure 4.4: Average precision, recall and $g$-mean results for the tested methods.](image-url)
Table 4.1: Detailed precision (P), recall (R) and g-mean (G) results obtained by applying 6 different methods to 10 images of ancient music.

<table>
<thead>
<tr>
<th>Image</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Niblack</td>
<td>88.02</td>
<td>83.63</td>
<td>48.64</td>
<td>83.83</td>
<td>75.24</td>
<td>69.09</td>
<td>91.03</td>
<td>81.95</td>
<td>91.12</td>
<td>95.28</td>
<td>80.78</td>
</tr>
<tr>
<td>R</td>
<td>91.99</td>
<td>80.18</td>
<td>82.83</td>
<td>84.42</td>
<td>82.32</td>
<td>88.77</td>
<td>71.86</td>
<td>80.30</td>
<td>81.68</td>
<td>89.81</td>
<td>83.42</td>
</tr>
<tr>
<td>G</td>
<td>89.99</td>
<td>81.89</td>
<td>63.47</td>
<td>84.12</td>
<td>78.70</td>
<td>78.31</td>
<td>80.88</td>
<td>81.12</td>
<td>86.27</td>
<td>92.51</td>
<td>81.73</td>
</tr>
<tr>
<td>Sauvola</td>
<td>92.03</td>
<td>87.01</td>
<td>94.51</td>
<td>92.47</td>
<td>81.03</td>
<td>94.01</td>
<td>90.04</td>
<td>84.02</td>
<td>89.40</td>
<td>96.50</td>
<td>90.10</td>
</tr>
<tr>
<td>R</td>
<td>91.17</td>
<td>81.37</td>
<td>93.17</td>
<td>85.56</td>
<td>79.32</td>
<td>92.22</td>
<td>80.33</td>
<td>83.31</td>
<td>89.08</td>
<td>94.33</td>
<td>86.99</td>
</tr>
<tr>
<td>G</td>
<td>91.60</td>
<td>84.14</td>
<td>93.84</td>
<td>88.95</td>
<td>80.17</td>
<td>93.11</td>
<td>85.05</td>
<td>83.66</td>
<td>89.24</td>
<td>95.41</td>
<td>88.52</td>
</tr>
<tr>
<td>Otsu</td>
<td>70.16</td>
<td>73.01</td>
<td>24.12</td>
<td>65.69</td>
<td>67.25</td>
<td>69.87</td>
<td>83.93</td>
<td>62.60</td>
<td>79.74</td>
<td>85.92</td>
<td>68.23</td>
</tr>
<tr>
<td>R</td>
<td>90.72</td>
<td>85.53</td>
<td>53.78</td>
<td>81.91</td>
<td>85.11</td>
<td>86.10</td>
<td>63.98</td>
<td>78.85</td>
<td>80.71</td>
<td>87.06</td>
<td>79.35</td>
</tr>
<tr>
<td>G</td>
<td>79.78</td>
<td>79.02</td>
<td>36.01</td>
<td>73.35</td>
<td>75.65</td>
<td>77.57</td>
<td>73.28</td>
<td>70.17</td>
<td>80.22</td>
<td>86.49</td>
<td>73.15</td>
</tr>
<tr>
<td>Fuzzy C-Means</td>
<td>95.65</td>
<td>95.76</td>
<td>91.47</td>
<td>94.81</td>
<td>79.23</td>
<td>98.16</td>
<td>97.33</td>
<td>86.74</td>
<td>98.50</td>
<td>99.01</td>
<td>93.67</td>
</tr>
<tr>
<td>R</td>
<td>74.57</td>
<td>68.45</td>
<td>75.20</td>
<td>72.36</td>
<td>60.96</td>
<td>53.37</td>
<td>46.65</td>
<td>64.00</td>
<td>46.23</td>
<td>81.43</td>
<td>64.32</td>
</tr>
<tr>
<td>G</td>
<td>84.45</td>
<td>80.96</td>
<td>82.93</td>
<td>82.83</td>
<td>69.49</td>
<td>72.38</td>
<td>67.38</td>
<td>74.51</td>
<td>67.48</td>
<td>89.79</td>
<td>77.22</td>
</tr>
<tr>
<td>Canny Niblack</td>
<td>90.30</td>
<td>83.38</td>
<td>45.36</td>
<td>91.24</td>
<td>76.24</td>
<td>74.82</td>
<td>91.56</td>
<td>82.12</td>
<td>91.25</td>
<td>98.00</td>
<td>82.43</td>
</tr>
<tr>
<td>R</td>
<td>91.36</td>
<td>80.26</td>
<td>62.64</td>
<td>84.59</td>
<td>82.70</td>
<td>88.57</td>
<td>71.76</td>
<td>80.43</td>
<td>81.24</td>
<td>90.11</td>
<td>81.37</td>
</tr>
<tr>
<td>G</td>
<td>90.83</td>
<td>81.81</td>
<td>53.30</td>
<td>87.85</td>
<td>79.41</td>
<td>81.41</td>
<td>81.05</td>
<td>81.27</td>
<td>86.10</td>
<td>93.97</td>
<td>81.70</td>
</tr>
<tr>
<td>Canny Sauvola</td>
<td>91.98</td>
<td>87.03</td>
<td>93.31</td>
<td>92.56</td>
<td>79.92</td>
<td>91.90</td>
<td>90.16</td>
<td>80.10</td>
<td>89.20</td>
<td>96.50</td>
<td>89.27</td>
</tr>
<tr>
<td>R</td>
<td>88.78</td>
<td>80.55</td>
<td>92.16</td>
<td>83.70</td>
<td>77.09</td>
<td>89.67</td>
<td>78.30</td>
<td>80.59</td>
<td>86.23</td>
<td>94.47</td>
<td>85.15</td>
</tr>
<tr>
<td>G</td>
<td>90.37</td>
<td>83.73</td>
<td>92.73</td>
<td>88.02</td>
<td>78.49</td>
<td>90.78</td>
<td>84.02</td>
<td>80.34</td>
<td>87.70</td>
<td>95.48</td>
<td>87.17</td>
</tr>
</tbody>
</table>

Figure 4.5: G-mean results for the tested methods, in each of the 10 images.
servable in the results created by the methods of Otsu, Niblack and Canny with Niblack post-processing. The Fuzzy C-Means method was particularly aggressive with this image, with part of the staff lines getting erased. More varying results are observable in image 3. The Otsu and Fuzzy C-Means methods performed very poorly, showing big areas of degradation. Niblack’s method, either when used alone or as post-processing for the Canny edge detection, also performed poorly, showing sparse but still large interference regions. Sauvola’s method, either after edge detection or used alone, was able to remove almost all degradation. No method was able to remove the bleed-through pixels, which appear to the right in image 3. This is due to the high color intensity that bleed-through objects can appear with.

4.4 Summary

In the overall, Sauvola’s adaptive thresholding method was the one to produce the best results. Other methods provided median results, even good with some images, but had a more irregular behavior. It is therefore safe to assume that Sauvola’s thresholding method is a very good option to perform background homogenization.

There are, however, remaining limitations with restoration by background homogenization, mostly in what concerns bleed-through detection and removal. This kind of deterioration requires a specialized treatment and is the focus of Chap. 5.
Figure 4.6: Result of processing two of the original images with the chosen methods. Columns 1 and 2 correspond to the processing of images 2 and 3, respectively. Presented from top to bottom are: the original images; the result of binarizing the original images with Niblack thresholding, Sauvola thresholding, Otsu thresholding, Canny edge detection with Niblack post-processing, Canny edge detection with Sauvola post-processing, and Fuzzy C-Means; and the restored images using Sauvola thresholding, the method that performed the best.
5 Bleed-through Removal

This chapter considers approaches to restore the highly damaging deterioration characteristic of bleed-through. As was observed in Chap. 4, general restoration methods are not able to successfully deal with the problem of bleed-through. This has mostly to do with the fact that bleed-through interference often assumes very dark intensity levels, being difficult to recognize.

Current approaches for bleed-through removal are split between blind and non-blind methods. Blind methods process a single side of a leaf of paper – the recto – while non-blind methods use both sides – the recto and the verso.

More recto-based methods seem to exist, when compared to those that use both the recto and the verso. The principal advantage of using only the recto is naturally that one doesn’t require the verso to be present. There are times when the verso is not available so only recto-based methods are applicable. It may also be more efficient, in general, to use methods based only on the recto, although this highly depends on the actual method under consideration. However, while part of the degradation occurs independently on the two sides of a document, bleed-through usually provokes degradation that is highly dependent between the two sides. By not using the verso, recto-based methods have to use alternative features to detect which parts of the recto correspond to the verso.

On the other hand, restoration performed using both sides has the advantage of producing good results where recto-based restoration misses features to correctly distinguish interference from the valid image contents. As discussed in Chap. 4, background homogenization, which uses only the recto, is unable to successfully deal with the problem of bleed-through, as none of the experimented methods were able to remove it. Using the verso along with the recto, it becomes possible to trace the location of the recto bleed-through interference back to its origin, in the verso. However, double-sided methods are also usually more inefficient, although this once again depends on the particular method being considered.

5.1 State of the Art

Hereby presented are known methods related to the problem of bleed-through removal. Most of these methods are focused on images of text documents, while others also include documents with illustrations.

Recto-based Methods

Basic thresholding methods assume separable gray scale levels or distinctive features between the foreground and the background. Since the darkness of bleed-through may be comparable to or even greater than that of the valid foreground, a simple thresholding operation typically cannot be used to detect bleed-through. Therefore, the existing methods that target on the removal of bleed-through typically use additional features like the orientation of characters and the sharpness of the edges found within those images.

A method proposed by Tan et al [TCWS00, TCS+00, CTWS00] considers the extraction of a clean foreground from historical handwritten documents. It consists on adopting an edge detection algorithm together with an orientation filter to extract the foreground edges and remove the reverse side edges. Due to the property of the paper material of archival documents, they observe that, on the recto, the edges of strokes that sipped from the verso are not as sharp as those of the true foreground. This allows for the adoption of an edge detection algorithm to suppress unwanted interfering strokes. They also use the writing style as a contribution to filter out strong and persistent noisy strokes originating from the verso. This is based on the fact that the writing style is normally slanting from lower-left to upper-right. Their algorithm performs well and greatly improves the appearance of original documents. However, one problem with this method is mainly observed...
when the interference is so serious that the edges of the interfering strokes are even stronger than those of
the foreground. As a result, the edges of the interfering strokes will still remain in the restored images.

Nishida and Suzuki [NS03, NS02] adopt a multi-scale approach that employs a cyclic process that gener-
ates a restored image and its correction by detecting anomalies in comparison with the original image. This
approach assumes that the edge magnitudes of show-through components for texts and graphics are signif-
icantly smaller than those of the foreground. Therefore, it is possible to extract text and graphic components
on the recto through analysis of the edge magnitude distribution. In practice, the extraction is carried out by
thresholding the edge magnitude. The threshold depends on the physical properties of the paper sheet or the
type of document (e.g., magazine, journal, newspaper, advertisement). Therefore, the threshold is selected
by calculating statistics on optimal thresholds for sample images of each case. Furthermore, locally adaptive
binarization [Sha00] is applied to each color channel to make sure that characters and graphics are sepa-
rated. Their method starts with a preprocessing algorithm that mainly performs edge-preserving adaptive
smoothing [NM79] to remove noise and to transform halftone patterns into continuous tones. An edge image
is generated by calculating the edge magnitude for each pixel of the image [Cum91, LC91] and afterwards
binarized by global thresholding. Text and graphics components such as line drawings are extracted by ap-
plying a locally adaptive binarization technique [SP00] to each RGB color channel. Background colors are
locally estimated through color thresholding to generate a restored image. For each horizontal scan line, runs
with a specified maximum length are constructed sequentially from left to right so that each run contains no
foreground pixels. Color clustering classifies the pixels of each run into two representative colors, and the
brightest color of the two is applied to all the pixels in the run. Furthermore, vertical runs are also applied to
the image in the same way. Finally, the restored image is adaptively corrected through multi-scale analysis
along with the comparison of edge distributions between the original and the restored image.

Use of directional wavelets to remove interfering strokes from images of historical handwritten documents was
proposed by Wang et al [WXTL03]. They make use of the writing style of documents to determine differences
between the orientations of foreground and interfering strokes. A directional wavelet transform exploits the
directional property of the strokes in order to separate the foreground strokes and the interference mainly in
different wavelet frequency domains. The foreground and the interfering strokes are slanting along the direc-
tions of 45 and 135 degrees, respectively. The directional aspect of the transform is capable of distinguishing
the foreground and reverse side strokes, effectively removing the interference. An image recovery step is also
used to recover foreground strokes that may become broken when interfering strokes that were intersecting
them are removed. This step is also used to recover small pieces of strokes that, having orientations ex-
tremely different from the majority, are removed together with the interference. Through the recovery step,
gray level pixels within a 7x7 neighborhood are recovered from the original document image. The size of the
neighborhood is based on the average width of the strokes in the documents.

Drira [Dri06] proposes a non-supervised segmentation method for ink bleed-through in document images.
This method recursively applies a K-Means algorithm on dimensionally reduced and decorrelated image data.
The dimension of an image is reduced and its data is decorrelated using principal component analysis com-
puted in the RGB color space. This improves the quality of the classification due to its properties which
reduce the data space and eliminate associations between data. The K-Means algorithm is then applied with
parameter $K = 2$, resulting in two classes of image pixels. The pixels of each class are back-projected into the
original color space and a decision is made about which class represents foreground elements. Logarithmic
histograms are computed for the two classes and the one with darker pixel values is used. This will be the
input of the next recursion step. Recursion is applied a specified number of times, determined empirically to
be equal to 3 to produce the best results. Finally, the valid foreground pixels are superimposed on the average
color of the detected background pixels.
Other approaches include tone correction replacing highlight colors with white through analysis of luminance distribution over the image [MU01], independent component analysis [TSMB04], self-organizing maps [SBH04] and color analysis [LBE04].

**Recto and Verso based Methods**

Typically, but not always, methods based on both the recto and the verso are comprised of:

- Matching of the recto and verso images.
- Heuristics to distinguish bleed-through from the valid foreground.
- Bleed-through removal.
- Inpainting [BSCB00], to fill-in the removed bleed-through regions.
- Pre or post-processing algorithms.

Matching of the recto and verso images is accomplished using registration, a fundamental task in image processing that consists on overlaying two or more images of the same scene taken at different times, from different viewpoints, or by different sensors. There are multiple surveys on image registration [Bro92, MV98, KS83, ZF03]. The necessity for using registration comes from the variability in the way the recto and verso images are captured. Although it is true that there exist image scanners which can process both sides of a leaf of paper at the same time, those machines do not account for the majority of the existing scanners and cannot be used with all kinds of documents. When scanning the recto and the verso using typical machines, many differences may arise in the resulting images, when compared to each other. This includes:

- Shifting – the documents may not be equally positioned.
- Rotation – the documents may not be equally aligned.
- Resolution – the documents may have been scanned at different resolutions.
- Geometrical transformation – the documents may show changes observable in the image as a whole; for instance, when thick books are scanned, it is common to observe this effect near the spine of the book.
- Illumination – documents may have been scanned with different light conditions or may present uneven illumination.

In order to match the recto with the verso it is therefore necessary to take all these conditions into account during the registration process. Some of the methods that will be described do account for most of these conditions, while others require a manual registration of the images.

Sharma [Sha00, Sha01] developed a simple model of the phenomenon of show-through effects and proposed an adaptive linear filtering scheme using scans of both sides of a document. He states that recovery of the show-through corrected image cannot be accurately done using the recto scan alone because it is not possible to reliably distinguish between light gray printing on the recto and low-contrast show-through from the verso. It is assumed that the distortion in the documents is due to show-through and that the impairment when scanning such documents can be well modeled by the properties of the physical scanning process. Show-through can then be canceled using adaptive filtering techniques. First, an approximate alignment for the recto and verso scans is determined by identifying the corresponding image features in the recto show-through and the scan of the verso. The reflectance of the white paper unprinted on both sides is estimated by averaging the reflectance values over a region of the scan that has no printing on either side. Using the estimate of the reflectance, scan data from the recto is converted to density relative to the paper white and data for the verso is converted to absorptance. The scanned image values for the recto and verso are then examined and compared to the paper white reflectance to determine if the local neighborhood about the current pixel location (and including it) contains any printing on either side. If the local neighborhood has printing on the verso but no printing on the recto, the filter coefficients are adapted. The processing is then repeated for
the next pixel location, continuing until the complete image has been processed. Since the desired output
is not known for regions with printing on the recto, the filter coefficients cannot be adapted in those regions.
No adaptation is performed for regions that have no printing on either side to keep the filter coefficients from
drifting due to the noise in these regions.

Dubois and Pathak [DP01] propose a method based on simultaneously processing both sides of a gray-level
manuscript using a six-parameter affine transformation to register the two sides. This method does not highly
depend on a model of the bleed-through process. Once the two sides have been correctly registered, areas
consisting primarily of ink bleed-through are identified using a thresholding technique and replaced by an
estimate of the background color or intensity. A pixel is classified as bleed-through in an area without recto
writing if its value on the verso exceeds a certain threshold and is sufficiently darker than its value on the
recto, according to a second threshold.

Another approach is described by Wang and Tan [WT01]. This method shares similarities with the recto-
based method, from the same authors, that was previously described [TCWS00, TCS’00, CTWS00], with
the difference that it now uses an additional pixel matching process to match the two sides of a document.
Background noise is earlier removed using Otsu’s global thresholding method [Ots79]. Mortagh’s point pattern
matching method [Mur92] is used to match the pixels, with the removal of the pixel with lower intensity from
every two correspondent pixels. This step leaves remaining fragments of partially unmatched interfering
strokes. Furthermore, some of the foreground strokes may become broken when interfering strokes that
were intersecting them are removed. Two properties are observed in the foreground strokes with respect
to the interfering strokes: 1) the edges of the foreground strokes are sharper than those of the interfering
strokes; and 2) the orientations of the foreground strokes and the interfering strokes are markedly different (the
foreground strokes are generally slanting at 45° with respect to the horizontal line, while the interfering strokes
are at 135°). These two properties are therefore incorporated into a modified Canny edge detector [Can86]
to further eliminate remaining interference that was not removed in the pixel matching process.

Cao et al [TCS02, CTS01] propose a wavelet reconstruction process applied to iteratively enhance the fore-
ground strokes and smear the interfering strokes. Doing so strengthens the discriminating capability of an
improved Canny edge detector [Can86] against the interfering strokes. A manual registration of the two sides
is used. The two sides of the document are combined into a single composite image for further process-
ing. The verso is inverted, flipped and added to the recto, resulting in a subtraction of the contents on the
verso. This weakens most of the interfering strokes but also affects some valid foreground strokes on the
recto. These foreground strokes, though somewhat impaired, now serve as seeds to start an enhancement
and smearing process. The idea is to detect the foreground strokes on the recto and enhance them using
wavelets. To detect the foreground strokes on either side an improved Canny edge detection algorithm with
an orientation filter and orientation constraint [CTWS00, TCS’00] is first adopted to pick up the edges of the
foreground strokes based on the observation that these are sharper than the interfering strokes. The orien-
tation filter favors foreground strokes that are predominantly slanting at a particular angle against interfering
strokes slanting at a different angle. The orientation constraint helps minimize erroneous edges caused by
nearby interfering strokes. The resultant edges then serve as loci to recover streaks of gray-level images of a
predetermined width from the foreground overlay images. The recovered streaks are binarized using Niblack’s
adaptive thresholding [Nib86].

Dubois and Dano [DD05] describe an improvement to the previous work of Dubois and Pathak [DP01], al-
ready described. They find the scheme of the previous method to introduce too many misclassification errors,
so a more elaborate algorithm was devised. They use registration with an optimization method based on an
affine transformation (that models shifting, rotation and even some skew). Each side of a document is seg-
mented into four regions: foreground only, background only, bleed-through only and mixed bleed-through and
foreground. Then, the areas identified as bleed-through-only are replaced with an estimate of the background using an inpainting technique [BSCB00]. They state that a pixel-by-pixel comparison between the verso and the recto may not work well, based on the observation that bleed-through may be somewhat diffused as it has sipped through the paper and can thus occupy a larger area on the reverse side. Thus, local windows were introduced into the algorithm. Bright areas of the document are considered to be background. Areas that are dark on the side of interest and not on the reverse side are considered to be foreground. Areas that are dark on both sides, but more so on the reverse side, are considered to be bleed-through, while if they are dark on both sides, but similar, they are considered regions of overlap.

**Applicability**

The recto-based methods described generally use the orientation of characters as a feature. Characters from the recto typically have a different orientation from those of the verso, so their orientation can be used as a distinguishing feature. However, this is not observable in images of ancient music. Among other elements, music notation of ancient music includes notes with square and diamond shapes (possibly including tails), staffs, barlines and rests. By observing multiple existent images one can conclude that these are typically written in a nearly vertical orientation, and therefore the described methods that base themselves on the orientation cannot be applied to the restoration of ancient music.

Another commonly used feature in recto-based methods is the magnitude of the edges on the recto. With the observation that the edges of the bleed-through regions are not as sharp on the recto as they are on the verso, some methods use this as a feature to detect bleed-through. This feature is, however, usually used by the described methods alongside the orientation feature, as a complement. The use of edge detection alone for restoration has already been tested in Chap. 4. As was demonstrated, not only edge detection was unable to successfully deal with the problem of bleed-through, it also was not the method that obtained the best results.

Methods based on both sides of a document are, on the other hand, supposedly capable of producing better results. It is not uncommon to have images of both the recto and the verso available, so there is certainly room for the applicability of these methods. The method by Sharma [Sha00, Sha01] does not apply to the case of bleed-through [DD05], due to its nature in the derivation of a physical model describing the show-through phenomenon that occurs when scanning documents. This model is based on properties of the physical scanning process, a condition that does not influence the appearance of interference in the case of bleed-through. The approach by Cao et al [TCS02, CTS01] also does not seem to fit the problem of ancient music restoration, as the recto and verso are used in a way that mostly subtracts the verso from the recto, after some simple transformations. This naturally removes valid parts of the foreground, so a reconstruction step is afterwards used to recover them. However, this reconstruction is based on the orientation of the characters which, once again, does not apply to the case of ancient music. The method described by Wang and Tan [WT01] shares similarities with this last one, in that the pixel matching process used removes, from two pixels corresponding to each side of a page, the one with the lower intensity. This is, once again, followed by a reconstruction which uses features that cannot be applied to the problem of ancient music.

Besides these methods, some remain which do seem to apply to the context of ancient music restoration. In addition to being devised to the problem of bleed-through, the methods of Dubois and Dano [DD05], and Dubois and Pathak [DP01], use more complex heuristics when it comes to the decision of which region each pixel belongs to. The former is stated as an improvement to the later.

Careful thought should also be given to a possible reconstruction phase. When using the described methods in an aggressive way, possibly necessary when dealing with cases of extreme degradation, there is a big probability that these methods will destroy valid bits of the foreground – probably the staff lines, which can be quite fragile. A reconstruction phase may be performed to try to recover the lost parts of the foreground.
5.2 Bleed-through Restoration Method

The bleed-through restoration method processes the images corresponding to the two sides of a leaf of paper in order to restore them. As outlined in Fig. 5.1, the method is comprised of multiple steps. First, the verso image is flipped horizontally in order for its coordinate system to match that of the recto. The recto and the verso images are then registered. Background homogenization is performed using the adaptive thresholding method that provided the best results in the experiments previously detailed in Chap. 4, i.e., Sauvola’s thresholding. Following thresholding, the staff lines are detected. Features are then extracted and classification performed in order to remove bleed-through, after which a final reconstruction step will generate the final restored image.

The feature extraction and classification step can be accomplished in many different ways, and therefore constitutes the core of the bleed-through restoration method. As such, multiple heuristics were researched and experimented, constituting the focus of Sect. 5.3.

The result of the bleed-through removal step is a binary image that distinguishes the foreground from the background. All the detected deterioration, including bleed-through, is incorporated into the background. Throughout the exposition of the method, the recto image will stand as the target for restoration, with the verso image being used, in addition, to restore it. The restoration of the verso image follows the exact same approach and will therefore not be covered here.

The bleed-through problem formulation is now described, and the restoration method presented afterwards.

Problem Formulation

The problem is formulated using a notation that is similar, in part, to that found in the work of Dubois and Dano [DD05]. Let $f_{or}(x, y)$ and $f_{ov}(x, y)$ denote the original recto and verso digital images, respectively, after being converted to gray scale and flipping the verso. The points $(x, y)$ lie on a two-dimensional rectangular space and the range of the functions is the interval $[0, 1]$, with 0 and 1 corresponding to white and black, respectively. Each of the two images is comprised of foreground, background and bleed-through areas. The foreground area contains the writing that was intentionally applied to the paper. The bleed-through area contains part of the verso’s writing that has bled through the page. The remainder of the recto corresponds to the background. The original recto image can therefore be represented as

$$f_{or}(x, y) = f_{orf}(x, y) + f_{orb}(x, y) + f_{orbg}(x, y) ,$$  (5.1)

where $f_{orf}(x, y)$, $f_{orb}(x, y)$ and $f_{orbg}(x, y)$ correspond to the original recto foreground, bleed-through and background areas, respectively.

This allows to model the relationship between the two sides:

$$f_{or}(x, y) = C(f_{ov}(x, y), f_{orb}(x, y), f_{orbg}(x, y)) ,$$  (5.2)

where $f_{ov}(x, y)$ denotes the original verso foreground and $C$ is a function that combines its arguments in some way. Possible models for this combining function were presented by Dubois and Pathak [DP01]. A simple additive model defines it as

$$C(\mu, \rho, \tau) = \mu + \rho + \alpha \cdot \tau ,$$  (5.3)

where $\alpha$ represents the attenuation of the verso foreground sipping to the recto.

Still, the bleed-through removal method here presented does not highly depend on a specific model of the bleed-through effect. Its ideal purpose is to create a restored recto image, which we denote as $f_r(x, y)$, that
Figure 5.1: Activity diagram for the bleed-through restoration method, including its input and output objects.
nullifies both the background and the bleed-through areas, maintaining only the valid writing:

\[ f_r(x, y) = C(f_{rbg}^{\text{orfg}}(x, y), 0, 0) = f_{rbg}^{\text{orfg}}(x, y) \ . \]  

(5.4)

Conceptually, the intent is to determine the parts of an image that correspond to the valid intentional writing. To this accomplishment, the background and bleed-through areas need to be removed. As the bleed-through areas depend on the verso’s writing, information from the verso needs to be incorporated and combined with the recto. In practice, however, it is not possible to establish a perfect relation between the two sides. On one hand, the sipping of ink is irregular, as only part of the verso writing does sip to the recto and the bleed-through area may be somewhat diffuse, thus occupying a larger area in the recto than it originally occupied in the verso. On the other hand, ink may have sipped to areas of the recto that already contained valid writing, thereby making it extremely difficult, even for human readers, to distinguish the foreground from bleed-through. Therefore, the main purpose of a bleed-through removal method is to restore as much deterioration as possible, recognizing the problem’s difficulties beforehand.

**Registration**

The necessity for using registration comes from the need to match the recto with the verso, as the two images most probably contain variations, including mostly shifting, rotation, and some skewing. These properties can be modeled by an affine transformation \( A_r \), of parameter vector \( t = [t_{11} \ t_{12} \ t_{13} \ t_{21} \ t_{22} \ t_{23}] \) [Bro92], defined as [DD05]

\[ (A_r f)(x, y) = f(t_{11} x + t_{12} y + t_{13}, t_{21} x + t_{22} y + t_{23}) \ . \]  

(5.5)

The parameter vector is estimated by solving the optimization problem:

\[ \hat{t} = \arg \min_t \sum_x \sum_y \left[ f_{\text{orfg}}^{\text{orfg}}(x, y) - (A_r f_{\text{orfg}}^{\text{orfg}})(x, y) \right]^2 \ . \]  

(5.6)

The registered verso image, i.e., the image that results after applying the affine transformation to the original verso, can then be calculated with

\[ f_{\text{vr}}^{\text{orfg}}(x, y) = (A_r f_{\text{orfg}}^{\text{orfg}})(x, y) \ . \]  

(5.7)

while the recto image remains unchanged, thereby having its registered version represented as

\[ f_{\text{vr}}^{\text{orfg}}(x, y) = f_{\text{orfg}}^{\text{orfg}}(x, y) \ . \]  

(5.8)

**Background Homogenization**

After registering the images, the registered recto background \( f_{\text{rbg}}^{\text{orfg}} \) is estimated and removed, along with the degradation it contains. Sauvola’s adaptive thresholding method [SP00] was used, as it was the background homogenization method to give the best results, in the experiments of Chap. 4. The method is not described here as it has already been described.

Let \( f_{\text{rbg}}^{\text{orfg}} \) denote the binary image that results after applying Sauvola’s thresholding to \( f_{\text{orfg}}^{\text{orfg}} \). This image contains all background pixels set to 0 and the remaining pixels set to 1. It will be the task of the classification step to distinguish from the registered recto foreground and bleed-through, now that the background has been detected.

**Staff Line Detection**

Staff lines are detected because line pixels may become incorrectly classified as bleed-through in the classification step, leading to broken lines. This allows to restore those lines as a post-processing step.
To this intent, the method of Pinto at al [PVdCS03] was used, as it has been shown to perform well with images of written ancient music. This method uses horizontal projections and small rotations of \( f^b_r \), finding peak areas of the projections and classifying them as staff lines.

**Feature Extraction and Classification**

[PVdCS03] This is the phase where multiple heuristics, to be experimented, vary. These experiments are detailed in Sect. 5.3 for the sake of clearness, as including them here could create a distraction from the focus of this section, i.e., the bleed-through restoration method.

Extraction of features is accomplished by taking both the recto \( f^a_r \) and the verso \( f^a_v \) into account. All the pixel positions \((x, y)\) are selected as possible candidates according to the following criteria:

\[
\forall_{x,y} \left( (f^b_r(x, y) = 1) \land (f^a_r(x, y) < f^a_v(x, y)) \implies ((x, y) \in \text{Candidates}) \right).
\]

From these candidates, different features are extracted by different methods in order to classify pixels as bleed-through or valid foreground. It should be noted that the classification of background pixels was already performed in the Background Homogenization phase.

**Post-processing**

Having classified all pixels as bleed-through or foreground, there are now \( f^a_{fr} \) and \( f^a_{fbt} \), where the first corresponds to the registered image without bleed-through. As a final step, two operations need to be performed. First, the initially detected staff lines are restored. Second, the bleed-through diffusion, visible as a set of pixels surrounding bleed-through, is suppressed. To this intent, a window of size \( 5 \times 5 \) is centered on each bleed-through pixel and all the pixels connected to it, inside that window, showing intensity levels at most 0.04 darker or lighter than the center pixel value, are marked as bleed-through. **expandir esta parte da difusão**

This produces the final restored image \( f_r \), containing the detected valid writing, which is derived from the detected foreground \( f^a_{fr} \), according to (5.4) and (5.8), as

\[
f_r(x, y) = f^a_{fr}(x, y) = f^a_{fbt}(x, y).
\]

5.3 **Feature Extraction and Classification Heuristics**

Different heuristics were used throughout the experiments, with the intent of determining which one provided the best results. These heuristics are naturally based on research on the field, but most of them are nonetheless considered original in the context of ancient music restoration, as they detect bleed-through in new and different ways. They are described next.

**Naive**

This is the most basic approach for bleed-through removal. No features are extracted and all bleed-through candidates are marked as bleed-through. This includes all pixels \((x, y)\) which have not been marked as background during Background Homogenization and have the verso color \( f^a_v(x, y) \) darker than the recto \( f^a_r(x, y) \).

**Minimum Difference**

This heuristic uses the differences between the pixels values of the recto and the verso as a feature, which is defined as

\[
diff(x, y) = f^a_v(x, y) - f^a_r(x, y) .
\]
The rationale for using this feature is that ink becomes lighter when sipping from the verso to the recto, therefore making it useful to have a measure of this intensity variation.

Taking the calculated differences into account, a mindiff threshold is used with each pixel, and all pixels \((x, y)\) such that

\[
diff(x, y) \geq \text{mindiff}
\]  \hspace{1cm} (5.11)

are marked as bleed-through. This threshold is set to a fraction of the range between maximum and minimum pixel values allowed. This range naturally depends on the storage space given for each color component so, for instance, what normally happens when using 1 byte to represent a color component is that the minimum and maximum pixel values allowed are 0 and 255, respectively, yielding the range of 255. From tests performed with mindiff ranging from \(\frac{\alpha}{30}\) to \(\frac{\alpha}{12}\), where \(\alpha\) is the range between the maximum and minimum pixel values allowed, the value of \(\text{mindiff} = \frac{\alpha}{10}\) provided the best results.

**Minimum Maximum Difference**

This heuristic extends the Minimum Difference heuristic and adds another threshold: maxdiff, the maximum difference, which denotes the maximum difference allowed for a pixel to be marked as bleed-through. The pixels \((x, y)\) are now marked as bleed-through if they obey the condition

\[
\left( \diff(x, y) \geq \text{mindiff} \right) \land \left( \diff(x, y) \leq \text{maxdiff} \right)
\]  \hspace{1cm} (5.12)

From tests performed with mindiff and maxdiff ranging from \(\frac{\alpha}{30}\) to \(\frac{\alpha}{12}\) and from \(\frac{\alpha}{23}\) to \(\frac{\alpha}{11}\), with maxdiff always greater than mindiff and where \(\alpha\) is as previously defined, the values of \(\text{mindiff} = \frac{\alpha}{10}\) and \(\text{maxdiff} = \frac{\alpha}{2}\) provided the best results.

**Difference Range**

This heuristic extends the Minimum Maximum Difference heuristic by automatically determining the mindiff threshold and using it to set the value of the maxdiff threshold according to a diffrange value.

The center of diffrange is determined to be the mean of the difference values of the possible bleed-through pixels, denoted as meandiff. This is based on the rationale that bleed-through pixels may be near the average difference between the recto and the verso pixels. As pixels are more and more away from the average difference, they become darker or lighter, hence possibly corresponding to the valid recto and background, respectively.

Having calculated the center of the difference range, the minimum and maximum difference thresholds are calculated as

\[
\text{mindiff} = \text{meandiff} - \text{diffrange}/2
\]  \hspace{1cm} (5.13)

\[
\text{maxdiff} = \text{meandiff} + \text{diffrange}/2
\]  \hspace{1cm} (5.14)

From tests performed with diffrange ranging from \(\frac{\alpha}{23}\) to \(\frac{\alpha}{25}\), the value of diffrange = \(\frac{\alpha}{23}\) provided the best results.

**Dark Minimum Difference**

This heuristic extends the Minimum Difference heuristic and strengthens its decision condition based on performing global thresholding on both the recto and the verso. Otsu’s thresholding method [Ots79] is applied to both sides, dividing them into foreground and background. A pixel \((x, y)\) is marked as bleed-through only if
its value on the verso is detected as foreground by Otsu’s thresholding and its value on the recto is not. The pixels \((x, y)\) are now marked as bleed-through if they obey the condition

\[
\text{diff}(x, y) \geq \text{mindiff} \land \left( f_{\text{Ot}}^r(x, y) = 0 \right) \land \left( f_{\text{Ot}}^v(x, y) = 1 \right) ,
\]

where \(f_{\text{Ot}}^r(x, y)\) and \(f_{\text{Ot}}^v(x, y)\) are the binary result of applying Otsu’s thresholding to the recto and the verso, respectively, with 0 corresponding to the background and 1 to the foreground pixels. This relates to saying that a pixel is marked as bleed-through if its verso is sufficiently dark (based on global thresholding), its recto is not sufficiently dark (again based on global thresholding), and they differ at least \text{mindiff} in their intensity values. The value of \text{mindiff} used here is the same as used in the Minimum Difference heuristic, i.e., \text{mindiff} = \frac{a}{16}.

### Thresholded Difference

This heuristic extends the Minimum Difference heuristic and automatically calculates its \text{mindiff} threshold by performing global thresholding on the difference values. Otsu’s global thresholding method \cite{Ots79} is once again used, thresholding the difference values with the rationale that the differences for bleed-through and valid foreground pixels may be observable as two peaks in the difference values histogram. This threshold value is then used as the minimum difference to mark a pixel as bleed-through.

### Connected Component Minimum Difference

This heuristic is characterized by performing decisions on groups of pixels, instead of on each individual pixel. The candidate pixels are grouped together and a decision if performed about whether the group is bleed-through.

Grouping of the pixels is performed by means of 4-connected components labeling of the binary image \(f_B^r(x, y)\). According to 4-connectivity, each foreground pixel is grouped with other foreground pixels if they are on the left, right, top or bottom of it.

Having labeled the pixels, a decision is performed for each group. The mean intensity value of the group pixels on the recto is compared to that on the verso, and those pixels are considered bleed-through if the mean values differ in at least \text{mindiff}, where \text{mindiff} means the same as in the Minimum Difference heuristic.

A group of pixels \(g\) is therefore marked as bleed-through if it obeys the condition

\[
\text{mean}^v(g) - \text{mean}^r(g) \geq \text{mindiff} ,
\]

where \(\text{mean}^v(g)\) and \(\text{mean}^r(g)\) correspond to the mean value of group \(g\) pixels on the verso and the recto, respectively.

The threshold \text{mindiff} is used with the same value as before, i.e., \text{mindiff} = \frac{a}{16}.

### Minimum Correlation

This heuristic uses the correlation between the pixels values of the recto and the verso as a feature. The correlation coefficients relate to information about whether the verso is similar to the recto, near a certain candidate pixel. For each candidate position, a window of size \(N \times N\) is centered on the recto and the verso, forming two matrices \(A\) and \(B\), respectively. The correlation coefficient for a given candidate is then calculated as

\[
r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{(\sum_m \sum_n (A_{mn} - \bar{A})^2)(\sum_m \sum_n (B_{mn} - \bar{B})^2)} ,
\]

where \(A_{mn}\) and \(B_{mn}\) are pixels with coordinates \((m, n)\) within the windows \(A\) and \(B\), respectively, and \(\bar{A}\) and \(\bar{B}\) are the means of the pixels within \(A\) and \(B\), respectively.
Taking the calculated correlations into account, a \( \text{mincorr} \) threshold is used with each pixel, and all pixels \((x, y)\) such that
\[
\text{corr}(x, y) = \text{mincorr}
\]
(5.18)
where \( \text{corr} \) is the correlation value, are marked as bleed-through.

From tests performed with \( N \) and \( \text{mincorr} \) ranging from 5 to 30 and from 0.2 to 0.8, the values of \( N = 15 \) and \( \text{mincorr} = 0.5 \) provided the best results.

**Minimum Difference and Correlation**

This heuristic is a composition of both the Minimum Difference and the Minimum Correlation heuristics. Its decision conditions are combined in order to create a stronger condition. The pixels \((x, y)\) are now marked as bleed-through if they obey the condition
\[
(\text{diff}(x, y) = \text{mindiff}) \land (\text{corr}(x, y) = \text{mincorr})
\]
(5.19)

The thresholds \( \text{mindiff} \) and \( \text{mincorr} \) are used with the same values that were used in the individual heuristics.

**Fuzzy Classification**

In this heuristic, 4 features are used: 1) recto intensity values; 2) verso intensity values; 3) correlation coefficients; and 4) recto and verso differences. These features were already introduced in the previously described heuristics.

In a classification problem the aim is to learn the behavior between the input and output of the training data. The use of fuzzy models in classification problems has been adopted in many domains [SR00, HP99, TA99], because they are able to solve difficult problems, exhibit robust behavior and present linguistic representations, which are easy to interpret.

The fuzzy models used here are an extension of the Takagi-Sugeno fuzzy models [TS85] in the affine form. This fuzzy classification rule is a fuzzy if-then rule whose consequent part is a class label [HP99, TA99]. It can be described by
\[
R^k : \text{If } x \in A^k \text{ then } x \in \text{class}^k \text{ with confident value } CV^k
\]
(5.20)
where \( k = 1, 2, \ldots, K \), \( i = 1, \ldots, K \), \( K \) denotes the number of rules in the rule base, \( R_i \) is the \( i \)th rule, \( n \) is the number of features, \( A_{i1}, \ldots, A_{in} \) are fuzzy sets defined in the antecedent space, \( y_i \) is the output feature for rule \( i \), \( a_i \) is a parameter vector and \( b_i \) is a scalar offset, \( \kappa \) is the number of classes, and \( CV^k \) is the confident value of the rule \( R^k \). The confident value of the if-then rule represents the rule weight interpreted as its confident strength. This type of model is used because it focuses on the precision of the obtained model.

To form the fuzzy system model from the data set with \( N \) data samples, given by \( X = [x_1, x_2, \ldots, x_N]^T \), \( Y = [y_1, y_2, \ldots, y_N]^T \) where each data sample has a dimension of \( n \) (\( N \gg n \)), first the structure is determined and afterwards the parameters of the structure are identified. The number of rules characterizes the structure of a fuzzy system. Fuzzy clustering in the Cartesian product-space \( X \times Y \) is applied to partition the training data. The partitions correspond to the characteristic regions where the system’s behavior is approximated by local linear models in the multidimensional space. Given the training data \( X_T \) and the number of clusters \( K \), a suitable clustering algorithm is applied. In this case, the Fuzzy C-Means [Bez81], one of the most widely used clustering algorithms, was adopted.

As a result of the clustering process, a fuzzy partition matrix \( U = [\mu_{ik}] \) is obtained. The fuzzy sets in the antecedent of the rules are identified by means of the partition matrix \( U \) which has dimensions \( [N \times K] \). One-dimensional fuzzy sets \( A_{ij} \) are obtained from the multidimensional fuzzy sets by projections onto the
space of the input variables $x_j$. This is expressed by the point-wise projection operator of the form $\mu_{A_{ij}}(x_{jk}) = \text{proj}_j(\mu_{ik})$. The point-wise defined fuzzy sets $A_{ij}$ are then approximated by appropriate parametric functions. The consequent parameters for each rule are obtained by means of linear least square estimation, which concludes the identification of the classification system.

5.4 Experiments

Experiments were conducted in order to determine which of the bleed-through heuristics achieves a better restoration. The method of Dubois and Dano [DD05], which was described in Sect. 5.1, was also tested. Once again, the quality of the restoration, as observed in the resultant images, was the main emphasis of the tests.

Methodology

A total of 14 images, i.e., 7 pairs of recto-verso images, scanned at a resolution of 150 dpi, were used throughout the experiments. Samples of these images are presented in Fig. 5.2. These images contain diverse degradation types, as well as different musical notations and illumination characteristics. They are representative of the majority of images present in the studied collections from the two referred libraries. All images were first manually restored, using graphics editing software, in order to allow them to be used later as a standpoint for comparison. From the 14 images, one side of each pair was used for training and the other for validation.

It must be noted that the method of Dubois and Dano [DD05] does not perform binarization. Its purpose is to detect and remove bleed-through areas, preserving the remaining parts of the image. Therefore, for it to be comparable with the heuristics previously detailed, Sauvola’s thresholding was applied to it. The image is thresholded after registration, but the algorithm proceeds as normal, ignoring the thresholded image. In the end, when bleed-through pixels have been detected, thresholds are combined to form the resulting image, which is therefore composed by the thresholded image with the detected bleed-through pixels removed.

The test images were processed by the chosen heuristics and compared to the manually restored images. The comparison follows the same approach of the experiments in Chap. 4, using the standard measures of precision, recall and geometric mean (g-mean). The description of these measures, as well as the way they were used in the calculation of the results, will not be described here as they were already described in the previous chapter.

Results

The results of evaluating the heuristics with the set of selected images are presented in Table 5.1. Detailed are the precision ($P$), recall ($R$) and g-mean ($G$) values, as well as their averages. It should be noted that there is an inherent degree of error in these results, as they are based on a pixel-wise comparison with manually restored images. When restoring those images by hand, it is hard to determine the exact class for each pixel, as the value of some pixels is not visually distinct.

The obtained average values can be easily visualized in the graph of Fig. 5.3. The MinDiff heuristic was the one to perform the best, obtaining the highest averages for both precision and recall, and therefore also g-mean measures.

According to the results performed by the heuristics, they can be divided into five groups: 1) MinDiff; 2) MinMaxDiff, DiffRange, DarkMinDiff and ThreshDiff; 3) CCMinDiff and Fuzzy; 4) MinDiffCorr and DuboisDano; and 5) Naive and MinCorr. This division makes sense for most of them, as heuristics of a group are usually based on similar features – although there are exceptions, as will be seen. The first group, consisting on one
Figure 5.2: Samples of the set of 14 images used throughout the tests.
Table 5.1: Detailed precision (P), recall (R) and g-mean (G) results obtained by applying 11 different heuristics to 7 pairs of images of ancient music.

<table>
<thead>
<tr>
<th>Image</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>20.75</td>
</tr>
<tr>
<td>P</td>
<td>10.20</td>
<td>27.69</td>
<td>6.17</td>
<td>46.47</td>
<td>23.52</td>
<td>12.85</td>
<td>18.32</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>64.38</td>
<td>57.09</td>
<td>41.25</td>
<td>57.50</td>
<td>40.01</td>
<td>58.93</td>
<td>63.21</td>
<td>54.62</td>
</tr>
<tr>
<td>G</td>
<td>25.62</td>
<td>39.76</td>
<td>15.95</td>
<td>51.69</td>
<td>30.68</td>
<td>27.51</td>
<td>34.03</td>
<td>32.18</td>
</tr>
<tr>
<td>MinDiff</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>74.00</td>
</tr>
<tr>
<td>P</td>
<td>55.52</td>
<td>74.36</td>
<td>76.15</td>
<td>78.94</td>
<td>68.47</td>
<td>78.46</td>
<td>86.11</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>99.61</td>
<td>91.44</td>
<td>90.69</td>
<td>97.02</td>
<td>81.59</td>
<td>93.30</td>
<td>97.60</td>
<td>93.04</td>
</tr>
<tr>
<td>G</td>
<td>74.36</td>
<td>82.46</td>
<td>83.10</td>
<td>87.52</td>
<td>74.74</td>
<td>85.56</td>
<td>91.68</td>
<td>82.77</td>
</tr>
<tr>
<td>MinMaxDiff</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>60.39</td>
</tr>
<tr>
<td>P</td>
<td>45.22</td>
<td>64.50</td>
<td>41.10</td>
<td>88.91</td>
<td>58.47</td>
<td>58.36</td>
<td>66.15</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>87.36</td>
<td>80.73</td>
<td>65.08</td>
<td>76.69</td>
<td>41.59</td>
<td>81.77</td>
<td>87.57</td>
<td>74.40</td>
</tr>
<tr>
<td>G</td>
<td>62.85</td>
<td>72.16</td>
<td>51.72</td>
<td>82.58</td>
<td>49.31</td>
<td>69.08</td>
<td>76.11</td>
<td>66.26</td>
</tr>
<tr>
<td>DiffRange</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>62.96</td>
</tr>
<tr>
<td>P</td>
<td>53.51</td>
<td>64.81</td>
<td>50.92</td>
<td>88.80</td>
<td>68.19</td>
<td>58.38</td>
<td>56.08</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>73.88</td>
<td>76.08</td>
<td>71.18</td>
<td>84.72</td>
<td>70.76</td>
<td>77.86</td>
<td>84.97</td>
<td>77.07</td>
</tr>
<tr>
<td>G</td>
<td>62.97</td>
<td>70.22</td>
<td>60.20</td>
<td>86.73</td>
<td>69.48</td>
<td>67.42</td>
<td>69.03</td>
<td>69.42</td>
</tr>
<tr>
<td>DarkMinDiff</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>59.00</td>
</tr>
<tr>
<td>P</td>
<td>45.52</td>
<td>64.36</td>
<td>41.15</td>
<td>88.94</td>
<td>58.47</td>
<td>58.46</td>
<td>56.11</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>89.61</td>
<td>81.44</td>
<td>65.69</td>
<td>77.02</td>
<td>81.59</td>
<td>83.30</td>
<td>87.60</td>
<td>80.89</td>
</tr>
<tr>
<td>G</td>
<td>63.86</td>
<td>72.40</td>
<td>51.99</td>
<td>82.77</td>
<td>69.07</td>
<td>69.78</td>
<td>70.11</td>
<td>68.57</td>
</tr>
<tr>
<td>ThreshDiff</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>61.64</td>
</tr>
<tr>
<td>P</td>
<td>55.98</td>
<td>67.57</td>
<td>45.24</td>
<td>77.71</td>
<td>61.47</td>
<td>54.91</td>
<td>68.60</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>92.30</td>
<td>88.69</td>
<td>80.66</td>
<td>80.18</td>
<td>90.62</td>
<td>77.27</td>
<td>78.28</td>
<td>84.00</td>
</tr>
<tr>
<td>G</td>
<td>71.88</td>
<td>77.41</td>
<td>60.41</td>
<td>78.94</td>
<td>74.64</td>
<td>65.14</td>
<td>73.28</td>
<td>71.67</td>
</tr>
<tr>
<td>CCMdiff</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>38.61</td>
</tr>
<tr>
<td>P</td>
<td>26.75</td>
<td>44.96</td>
<td>21.11</td>
<td>71.30</td>
<td>43.14</td>
<td>28.55</td>
<td>34.47</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>91.29</td>
<td>83.68</td>
<td>66.26</td>
<td>86.31</td>
<td>52.53</td>
<td>87.60</td>
<td>91.72</td>
<td>79.91</td>
</tr>
<tr>
<td>G</td>
<td>49.42</td>
<td>61.33</td>
<td>37.40</td>
<td>78.45</td>
<td>47.60</td>
<td>50.01</td>
<td>56.23</td>
<td>54.35</td>
</tr>
<tr>
<td>MinCorr</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>28.63</td>
</tr>
<tr>
<td>P</td>
<td>14.22</td>
<td>51.48</td>
<td>10.66</td>
<td>55.89</td>
<td>26.78</td>
<td>20.63</td>
<td>20.54</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>23.09</td>
<td>20.32</td>
<td>12.12</td>
<td>50.88</td>
<td>16.72</td>
<td>47.86</td>
<td>59.90</td>
<td>32.98</td>
</tr>
<tr>
<td>G</td>
<td>18.24</td>
<td>32.35</td>
<td>11.37</td>
<td>53.33</td>
<td>21.16</td>
<td>31.42</td>
<td>35.08</td>
<td>28.99</td>
</tr>
<tr>
<td>MinDiffCorr</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>37.97</td>
</tr>
<tr>
<td>P</td>
<td>21.55</td>
<td>62.30</td>
<td>21.16</td>
<td>68.50</td>
<td>34.06</td>
<td>28.80</td>
<td>29.46</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>51.69</td>
<td>47.61</td>
<td>21.18</td>
<td>70.71</td>
<td>21.05</td>
<td>73.19</td>
<td>84.80</td>
<td>52.89</td>
</tr>
<tr>
<td>G</td>
<td>33.37</td>
<td>54.46</td>
<td>21.17</td>
<td>69.60</td>
<td>26.78</td>
<td>45.91</td>
<td>49.98</td>
<td>43.04</td>
</tr>
<tr>
<td>Fuzzy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>44.39</td>
</tr>
<tr>
<td>P</td>
<td>24.90</td>
<td>71.90</td>
<td>48.70</td>
<td>71.30</td>
<td>51.40</td>
<td>40.00</td>
<td>2.50</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>92.10</td>
<td>81.40</td>
<td>71.80</td>
<td>77.00</td>
<td>79.50</td>
<td>83.30</td>
<td>78.80</td>
<td>80.56</td>
</tr>
<tr>
<td>G</td>
<td>47.89</td>
<td>76.50</td>
<td>59.13</td>
<td>74.10</td>
<td>63.92</td>
<td>57.72</td>
<td>14.04</td>
<td>56.19</td>
</tr>
<tr>
<td>DuboisDano</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>28.31</td>
</tr>
<tr>
<td>P</td>
<td>13.50</td>
<td>42.30</td>
<td>25.40</td>
<td>66.70</td>
<td>23.00</td>
<td>24.10</td>
<td>3.20</td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>83.20</td>
<td>73.70</td>
<td>50.90</td>
<td>80.70</td>
<td>85.90</td>
<td>84.50</td>
<td>89.40</td>
<td>78.33</td>
</tr>
<tr>
<td>G</td>
<td>33.51</td>
<td>55.83</td>
<td>35.96</td>
<td>73.37</td>
<td>44.45</td>
<td>45.13</td>
<td>16.91</td>
<td>43.59</td>
</tr>
</tbody>
</table>
heuristic only, obtained not only the best measured performance but also the most visually appealing results. More interestingly, it is not a complicated heuristic and is actually the base of many of the described heuristics. This demonstrates that the added complexities of the heuristics that derive from the MinDiff heuristic are unworthy. Most these heuristics are incorporated into group 2. Their results are very similar from each other, with ThreshDiff providing the best results. The fact that their results are worse than those of MinDiff is probably because they strengthen MinDiff’s decision conditions, thus not detecting as many bleed-through pixels. While not detecting as much bleed-through, they also do not correctly detect as many valid foreground pixels, hence providing worse restoration results. These heuristics can be seen as more conservative than MinDiff. Group 3 is interesting in the way that there does not seem to be a relation between the two heuristics, yet they provide similar results. While CCMinDiff uses the labels and mean differences of pixels in those labels as features, the Fuzzy Classification heuristic uses the difference of each pixel, the intensities and the correlation between pixels instead. Therefore, the similarity between the heuristics does not seem to be related in terms of the mechanism behind each heuristic. Group 4 includes the MinDiffCorr and DuboisDano heuristics. While their g-mean average is similar, the later provides a better recall average, while the former provides a better precision average. This means DuboisDano identifies more bleed-through pixels than MinDiffCorr, and therefore while detecting more valid bleed-through pixels as bleed-through it is also detecting more foreground pixels as bleed-through, hence having a lower precision. The last group includes two unrelated heuristics. The Naive heuristic provided overall bad results, as expected, which shows the need for having developed more specialized heuristics for restoring bleed-through. The MinCorr heuristic, however, performed surprisingly bad, having a slightly greater precision than the Naive heuristic at the expense of a lower recall average. This demonstrates that the use of correlation alone as a feature is not sufficient for a good restoration. As can be seen, this heuristic was improved when the pixel differences were added as a feature, in the MinDiffCorr heuristic. Still, the best heuristic using correlation as a feature was Fuzzy Classification, not reaching however the high averages that the pixel differences allowed for, when used as a feature in the MinDiff heuristic.

The g-mean scores for the 7 selected image pairs are represented in the graph of Fig. 5.4. This allows to describe the regularity of the heuristics from image to image. Once again, it should be noted that even though a line graph was used, the data is discrete and, as such, the lines connecting the data measured for each image only appear to ease the observation of which points belong to the same heuristic, besides the visualization of the regularity in the measured values. As can be observed, the MinDiff heuristic outperformed the other heuristics in all images. While most heuristics varied very greatly from image to image, the MinDiff heuristic was the most regular one, with ThreshDiff also providing a great regularity. This allows to mark these two heuristics as those to provide more predictable results when used with different images. Even with the image pair number 3, where most heuristics their lowest scores, MinDiff was able to obtain a good G-mean average.

Two of the images used within the experiments are presented in Fig. 5.5, along with the results of processing them with some of the chosen heuristics. Recalling the five groups of heuristics previously established based on the measured results, heuristics of the same group did provide similar output images. As such, Fig. 5.5 shows the results of the heuristics chosen to be illustrative of group they are in: 1) MinDiff; 2) ThreshDiff; 3) CCMinDiff; 4) DuboisDano; and 5) MinCorr. Overall, what distinguishes a good heuristic from a bad heuristic is the capability to remove bleed-through while maintaining most of the valid foreground intact. Some heuristics do remove bleed-through, but do so by fully erasing entire notes too, being quite damaging. Therefore, it is preferable to keep some bleed-through, preferably disperse, while maintaining the notes intact.

It can be noted that the difference between the heuristics that represent the best three groups, i.e., MinDiff, ThreshDiff and CCMinDiff, is subtle, which comes with accordance to the results that were previously measured. This difference gets greater when comparing these heuristics with those that represent the other two groups, i.e., DuboisDano and MinCorr. The readability is greatly improved with the removal of bleed-through
Figure 5.3: Average precision, recall and g-mean results for the tested heuristics.

Figure 5.4: G-mean results for the tested heuristics, in each of the 7 image pairs.
areas by the MinDiff heuristic, as well as by the ThreshDiff and CCMinDiff heuristics to some extent. The removal is naturally not perfect, as some bits of diffusion still remain. However, these bits are dispersed, therefore not creating a serious visual impact to the reader.

5.5 Summary

In the overall, the developed bleed-through restoration method was able to produce good results when using the Minimum Difference heuristic in the feature extraction and classification step. This was the heuristic to perform the best, from experiments performed with 10 developed heuristics, as well as a method which already existed in the literature – that of Dubois and Dano [DD05]. The developed heuristics vary in terms of the features they use, including the difference between recto and verso images, the correlation between windows on both sides, the intensities values of the pixels, and the set of connected components within the images. The operations performed by the heuristics using this features also vary, including simple operations that require minimum values of some features, to more complex operations like thresholding the difference between images and performing fuzzy clustering using multiple features. Part of the tested heuristics provided average results, but it is safe to assume that the Minimum Difference heuristic is a very good option to use in the method for bleed-through removal.
Figure 5.5: Result of processing two of the original images with the chosen heuristics. Columns 1 and 2 correspond to the processing of images 2 and 4, respectively. Presented from top to bottom are the original images and the result of removing bleed-through using the MinDiff, ThreshDiff, CCMinDiff, DuboisDano and MinCorr heuristics, respectively.
6 Software

The software produced throughout this thesis fulfilled both a goal and a means to achieve that goal. As stated before, one of the goals of this work was to create a restoration software to be used in the Portuguese National Library. On the other hand, in order to produce such software, experiments had to be conducted and prototyping software developed. As such, the produced software can be divided into two parts. First, methods were developed under the MATLAB environment and programming language. These methods were used throughout the experiments and lastly turned into a toolbox. When all the necessary experimentation had been conducted, a final library was developed in Java, for use in the Portuguese National Library. These two developed components are the focus of this chapter.

6.1 Restoration toolbox

The methods used in the experiments were developed in the MATLAB environment and programming language. The decision to use this environment was due to many factors:

- The support for quick prototyping. While testing methods, the focus is turned to development time, instead of execution time. It is a benefit to have reduced development time even if the resulting application is slower to execute. As such, a compromise has to be made regarding development and execution. Moreover, it is preferable to prototype in a language which provides an interpreted environment, in such a way that reduces the time from making changes to the prototype and watching the results of those changes. MATLAB provides such an environment and allows to work interactively while developing prototypes.

- The support for image processing. MATLAB provides a toolbox dedicated to image processing including the following features that were necessary for the experiments:
  - Importing and exporting images
  - Image enhancement, including linear and nonlinear filtering, filter design, deblurring, and automatic contrast enhancement
  - Image analysis, including line detection, mathematical morphology, edge detection, segmentation, region-of-interest (ROI) processing, and feature measurement
  - Color image processing, including color space conversions
  - Spatial transformations and image registration
  - Image visualization

- Its use is common practice in the research carried out in the Department of Mechanical Engineering, IST. Using MATLAB therefore provides a better intra-departmental communication and code reuse.

The developed MATLAB toolbox contains functions that are as parameterized as possible, and can therefore be also used in contexts beyond that of ancient music. Here is provided a summarized description of those functions. Complete documentation is found in the source code of the developed functions, which makes use of the default MATLAB help system. The methods for background homogenization and bleed-through removal are summarized in Tables 6.1 and 6.2, respectively, along with their signature.

6.2 The Complete Restoration Method

The experiments performed culminated in the specification of a complete restoration method and its subsequent implementation. This method was implemented as a software library to be used in the Portuguese
Table 6.1: Signature and summary of the background homogenization functions contained within the toolbox.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Function Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>img_bw = bhniblack (img, N, W)</td>
<td>Performs background homogenization using the Niblack method.</td>
</tr>
<tr>
<td>img_bw = bhsauvola (img, N, W)</td>
<td>Performs background homogenization using the Sauvola method.</td>
</tr>
<tr>
<td>img_bw = bhotsu (img)</td>
<td>Performs background homogenization using the Otsu method.</td>
</tr>
<tr>
<td>[img_bw, iters] = bhfuzzyclustering (img, N, M)</td>
<td>Performs background homogenization using the Fuzzy Clustering method.</td>
</tr>
<tr>
<td>[img_bw, img_bw_edges] = bhcannysauvola (img, N, sN, sW)</td>
<td>Performs background homogenization using the Canny Sauvola method.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Utility functions</th>
<th>Function Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[img_dilated, img_bw_dilated, img_bw_edges] = bhcannydilated (img, N)</td>
<td>Performs a morphology dilation operation after using Canny edge detection.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Testing functions</th>
<th>Function Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[T, TNP, TDP, CDP, P, R, F1, GM] = bhmeasureresult (detected_bw, valid_bw)</td>
<td>Measures an image after background homogenization, comparing it to a manually restored image and calculating the total number of pixels, total valid pixels, total detected pixels, correctly detected pixels, precision, recall, F1 and G-mean values.</td>
</tr>
<tr>
<td>bhmeasure</td>
<td>Automates the process of measuring multiple background homogenization results by comparing them to manually restored images.</td>
</tr>
<tr>
<td>bhtest</td>
<td>Automates the testing of the background homogenization methods with multiple parameter combinations.</td>
</tr>
</tbody>
</table>

National Library. A simple graphical interface was also developed with the purpose of testing, but due to its simplicity it will not be described here.

The Portuguese National Library presented a fairly flexible environment under which the software library was to be used and developed. Moreover, the greatest outcome of this thesis was the complete restoration method which was developed after experimentation of numerous methods and possibilities. As such, a detailed requirements document was not created as it would be overkill to the reduced set of requirements inherent to the developed software library. This includes:

- Given degraded images as input, the method should produce restored versions of those images.
- Emphasis on the restoration quality. The restoration method should produce images that are identifiably better than the original images. The comparison between images, in order to conclude whether the method achieves a good quality, should be conducted by experts and automatic methods known in the literature. The performance of the method is not, therefore, a priority.
- The software should be free of legal and commercial constraints, in order not to add to the expenses of the institution.

The software library was developed in the Java programming language. This was not a requirement, as multiple languages can currently coexist in the application environment of the institution. However, it is the language with which most of the software is currently developed, therefore allowing a better integration with the existing software without the need to glue different languages together.

As for the image processing functionality, multiple choices were considered. The image processing features considered necessary for this task were the same as those considered in Sect. 6.1. The following solutions were considered plausible alternatives:

1. Java Abstract Window Toolkit (AWT)
<table>
<thead>
<tr>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>imgr_res = btnaive (imgr, imgv, imgr_bw, imgv_bw)</code></td>
</tr>
<tr>
<td>Performs bleed-through removal using the Naive method.</td>
</tr>
<tr>
<td><code>imgr_res = btmindiff (imgr, imgv, imgr_bw, imgv_bw, min_diff)</code></td>
</tr>
<tr>
<td>Performs bleed-through removal using the Minimum Difference method.</td>
</tr>
<tr>
<td><code>imgr_res = btminmaxdiff (imgr, imgv, imgr_bw, imgv_bw, min_diff, max_diff)</code></td>
</tr>
<tr>
<td>Performs bleed-through removal using the Minimum Maximum Difference method.</td>
</tr>
<tr>
<td><code>imgr_res = btdiffrange (imgr, imgv, imgr_bw, imgv_bw, diff_range)</code></td>
</tr>
<tr>
<td>Performs bleed-through removal using the Difference Range method.</td>
</tr>
<tr>
<td><code>imgr_res = btdarkmindiff (imgr, imgv, imgr_bw, imgv_bw, min_diff)</code></td>
</tr>
<tr>
<td>Performs bleed-through removal using the Dark Minimum Difference method.</td>
</tr>
<tr>
<td><code>imgr_res = btthreshdiff (imgr, imgv, imgr_bw, imgv_bw)</code></td>
</tr>
<tr>
<td>Performs bleed-through removal using the Thresholded Difference method.</td>
</tr>
<tr>
<td><code>imgr_res = btccmindiff (imgr, imgv, imgr_bw, imgv_bw, min_diff)</code></td>
</tr>
<tr>
<td>Performs bleed-through removal using the Connected Component Minimum Difference method.</td>
</tr>
<tr>
<td><code>imgr_res = btmincorr (imgr, imgv, imgr_bw, imgv_bw, corr_w, min_corr)</code></td>
</tr>
<tr>
<td>Performs bleed-through removal using the Minimum Correlation method.</td>
</tr>
<tr>
<td><code>imgr_res = btmincorr (imgr, imgv, imgr_bw, imgv_bw, corr_w, min_corr)</code></td>
</tr>
<tr>
<td>Performs bleed-through removal using the Minimum Difference and Correlation method.</td>
</tr>
<tr>
<td><code>imgr_res = btduboisdano (imgr, imgv, imgr_bw, imgv_bw)</code></td>
</tr>
<tr>
<td>Performs bleed-through removal using the Dubois and Dano method.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Registration functions</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>imgr_res = btreg (imgr, imgv)</code></td>
</tr>
<tr>
<td>Registers two images.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Utility functions</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>img_res = btrmbg (img, mask)</code></td>
</tr>
<tr>
<td>Removes the background of an image and replaces it with a single color.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Testing functions</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>[N_N BT_BT N BT_BT P R F1 G] = btmeasureresult (detected_bw, valid_bw, btmask)</code></td>
</tr>
<tr>
<td>Measures an image after bleed-through removal, comparing it to a manually restored image and calculating the number of true positives, false positives, false negatives, true negatives, precision, recall, F1 and G-mean values.</td>
</tr>
<tr>
<td><code>btmeasure</code></td>
</tr>
<tr>
<td>Automates the process of measuring multiple bleed-through removal results by comparing them to manually restored images.</td>
</tr>
<tr>
<td><code>bttest</code></td>
</tr>
<tr>
<td>Automates the testing of bleed-through removal methods with multiple parameter combinations.</td>
</tr>
</tbody>
</table>
These libraries provide most of the image processing features required for the restoration method. The first five solutions were found to be the most complete and stable, being widely used currently. The first three, i.e., AWT, Java 2D and JAI, are all part of the standard Java platform. They represent the evolution of imaging in Java, from the reduced and simple initial set of operators provided by AWT, to JAI, a full featured, sophisticated and high performance library. JMagick is an open source Java interface for ImageMagick, an open source image processing software which can be used as a software library. OpenCV is an open source library aimed at computer vision and developed in C/C++. It was included in the list of plausible alternatives due to its current wide use in image processing applications.

Of these solutions, Java Advanced Imaging was chosen as the preferred one. The reasons for this choice were that it is part of the standard Java platform, it contains all the necessary image processing features, and is considered stable and mature. This provides some confidence in that the library will remain maintained and usable in the future.

The complete restoration method is presented in the activity diagram of Fig. 6.1. The steps of the method are described as activities, with their respective input and output objects which in this case correspond to state changes in the recto and verso images. The method begins with the decision of whether to perform restoration using both the recto and the verso, or just the recto, according to their availability. If only the recto is available, background homogenization is performed, after which the recto is reconstructed, reaching the restored state. If the verso is also available, besides the recto, the method goes through the steps of pre-processing, registration, background homogenization, staff line detection, bleed-through removal and, finally, reconstruction, where the recto reaches the restored state.

The registration step required the use of optimization functions which were not available in the Java API. To fulfill this absence, public domain Java optimization routines were used.
Figure 6.1: Activity diagram for the complete restoration method, including its input and output objects.
7 Conclusion

Written ancient music documents often present multiple types of degradation. Naturally, this degradation is visible in the images that result upon the digitization process performed in institutions like public libraries. This makes it harder to work with those images, not only from a human point of view – the readers and, therefore, the users of those images – but also from a machine point of view – processes that perform recognition and extract information from the images of ancient music documents.

Set out with well defined goals, the intent of this thesis was to research whether it was possible to automatically restore images of written ancient music, and to subsequently implement a restoration software to be used in the Portuguese National Library. To this achievement, multiple steps had to be taken. An observation was firstly carried out in order to generate a degree of acquaintance with the domain of this thesis – written ancient music. Doing so was necessary to determine which degradation problems images of ancient music usually portray and, therefore, establish an initial base for the concepts that needed to be studied. This corresponded to the study of the field of image segmentation, where all types of image segmentation, as presented in the literature, were analysed in detail. Having done so, there was the crucial need to inspect what was available in the literature that could be used for ancient music restoration. Due to the lack of surveys related to this context, all existing methods were researched and a state of the art survey created, which details existing methods that are related to binarization, document restoration and bleed-through removal. Research was then conducted in order to determine which of the available methods could be used for ancient music restoration, and to also study new ways of performing restoration. It became clear that there were two major problems to be solved: on one hand, the general restoration of an image of ancient music and, on the other hand, the removal of bleed-through, which required a specialized treatment. Having identified valid solutions for these problems, software was developed in conformance with the requirements of the Portuguese National Library.

The objectives of this thesis were therefore clearly met, having justified, developed and implemented a method that performs restoration of written ancient music.

Main Contributions

The greatest contribution of this thesis was the development of a new method for written ancient music restoration. This method is considered new, not only in its domain – ancient music – but also in the techniques it uses, as it provides a combination of techniques that were available in the literature as well as some new ideas that were developed. The method is flexible enough to be used in the majority of the situations found in institutions like public libraries. It also typically provides a good restoration quality, which was a desired requirement.

Another main contribution was the implementation of the restoration method as a software solution, in order to be used in the Portuguese National Library. This implementation followed the established requirements and was thoroughly tested. It is included in the compact disc that accompanies this dissertation.

In addition, there are two more contributions considered in this thesis. On one hand, the restoration toolbox that was developed and allows to use the multiple methods researched in this thesis not only with images of ancient music but also with other types of images, being of value to those that perform research in this domain. On the other hand, the state of the art survey that was conducted, which may serve as a standpoint for future studies in the field of ancient music restoration, as well as the more general field of document restoration.

Related Work

Related to this thesis was the publishing of two articles in international conferences:

- Methods for Written Ancient Music Restoration [CP07]
• Restoration of Double-sided Ancient Music Documents with Bleed-through [CAP]

Also related to this thesis was the participation in a state of the art article for the DIGMAP European project:

• State of the Art in Image Processing for Digitised Maps

**Future Work**

The work that should follow this thesis should be concerned, to some extent, with ways to improve the restoration method that was developed, and, to a greater extent, with ways to use this new method in other processes and systems.

The developed method may be subject to further improvement by performing research in a reduced set of areas which were not deeply considered during the conducted research. On one hand, this includes the field of mathematical morphology, which may be of help in the recognition of valid elements, within an ancient music image, in order to improve the final segmentation quality. On the other hand, methods that perform a mixture of edge detection and thresholding may also be researched. Even though all the conducted experiments with these kinds of methods indicate a certain degree of usability, there is the chance that they may be able to produce better results.

To a greater extent, as indicated, future work should be concerned with the applicability of the developed restoration method to processes existing in the systems related to document processing. Music recognition is one case of such processes. With the use of the restoration method, it is now possible to perform better at ancient music recognition. Even though work has already been done in the field of ancient music recognition [PVR’00, PVdCS03], not only could this work benefit from the method developed in this thesis, but also more research could and should be conducted in order to improve recognition methods. The use of restored documents, when taking the case of institutions like public libraries, should also be considered. The availability of these images to the public may be of great value, as they ease the users in their visualization task, so the use of these images should be considered in the processes that exist in these institutions.
Bibliography


[HK00] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000.


