Music Score Binarization Based On Content Knowledge

Telmo Tiago Barbosa Pinto

Master in Informatics and Computing Engineering

Supervisor: Jaime dos Santos Cardoso (Ph.D)

28th June, 2010
Music Score Binarization Based On Content Knowledge

Telmo Tiago Barbosa Pinto

Master in Informatics and Computing Engineering

Approved in oral examination by the committee:

Chair: Rui Camacho (Ph.D)
External Examiner: Manuel Ferreira (Ph.D)
Supervisor: Jaime dos Santos Cardoso (Ph.D)

31st July, 2010
Abstract

In Optical Music Recognition (OMR), a kind of Optical Character Recognition that focuses on the recognition of music symbols, it is useful to build a system that allows the transformation of digitalized music scores into a machine-readable symbolic format like MusicXML. These systems start processing the documents by searching for music staves. Many works have been done in the area, although one of the most important pre-processing steps that precedes the staff finding step has always been overlooked: the binarization.

This pre-processing consists in dividing the pixels of an image into two groups: the objects pixels (that get value 0) and the background pixels (that get value 1). Two major binarization techniques exist, the most common being the global thresholding in which a single intensity threshold is found for the whole image, dividing the pixels by intensity above or below that threshold. The adaptive methods, on the other hand, find a different threshold for each section of an image or even for each pixel.

Although many papers have been published describing methods for binarization of images, there is currently no study or comparison between such methods applied to music scores and no method developed that binarizes the documents with some a priori knowledge of the contents in a music score. Instead, most authors use an arbitrary method that in many cases results in loss of information, either by deleting the music staves or by covering the music symbols with noise.

This paper introduces Binarization based in LIne Spacing and Thickness (BLIST), a new method for content aware music score binarization that makes use of information that can be extracted from gray-scale music scores, before any pre-processing is applied. This method determines the staff line thickness and spacing in a gray-scale music score, and determines the threshold that assures the most staff lines with that thickness and spacing.

In addition to this global thresholding method, an adaptive version of BLIST is also described, by applying the method independently to vertical windows of the music score and interpolating the results to find a threshold for each vertical line of pixels.

These novel methods were compared to several state of the art binarization techniques, all applied to music scores, in order to analyze which algorithm assures the binarizations with the most staff lines and less noise. All the methods presented in this paper performed well above average. The global version of BLIST performed better than any other global method. The adaptive version of BLIST produced less noise but also seemed to result in more degraded staff lines.
Resumo

Em Reconhecimento Óptico de Música (Optical Music Recognition), uma área de Reconhecimento Óptico de Caracteres focada no reconhecimento de símbolos musicais, é útil a construção de um sistema que permita a conversão de de pautas digitalizadas para um formato simbólico que uma máquina seja capaz de ler, como MusicXML. Estes sistemas geralmente começam o processamento de documentos encontrando as linhas de pauta. Muitos trabalhos na área foram já publicados, embora um passo importante de pre-processamento que precede o reconhecimento de linhas seja geralmente ignorado: a binarização.

Este pré-processamento consiste em dividir os pixeis de uma imagem em dois grupos: os pixeis dos objectos (que recebem o valor 0) e os pixeis do background (que recebem o valor 1). Existem duas principais técnicas de binarização, sendo que a mais comum é o thresholding global no qual um único valor de intensidade limiar é encontrado para toda a imagem, dividindo os pixeis por intensidade acima ou abaixo desse limiar. Os métodos adaptativos, por outro lado, encontram um limiar diferente para cada secção da imagem, ou mesmo para cada pixel.

Embora muitas publicações tenham sido escritas descrevendo métodos de binarização de imagens, não existe actualmente um estudo ou comparação entre tais métodos aplicados a pautas de música, nem existe um método desenvolvido que utilize informação a priori sobre os conteúdos das pautas para as binarizar. Em vez disso, muitos autores utilizam métodos arbitrários que em muitos casos resultam em perda de informação, seja por apagarem as linhas de pauta ou por cobrirem os símbolos musicais com ruído.

Este trabalho, apresenta o Binarization based in Line Spacing and Thickness (BLIST), um novo método para binarização informada de pautas de música que utiliza conhecimento que pode ser extraído de pautas em gray-scale, antes de ser aplicado qualquer outro pré-processamento. Este método determina a espessura das linhas de pauta e distância entre elas, encontrando depois um limiar que garanta a presença de mais linhas com essa espessura e espaçamento.

Adicionalmente, uma versão adaptativa do BLIST é também descrita, aplicando o método independentemente a janelas verticais da pauta de música e interpolando os resultados para encontrar um limiar para cada linha vertical de pixeis.

Estes novos métodos foram ainda comparados a diversas técnicas de binarização do estado da arte, de modo a determinar que algoritmos asseguram binarizações com mais linhas de pauta e menos ruído. A versão global do BLIST teve um desempenho melhor que qualquer outro método global. A versão adaptativa do BLIST, apesar de produzir menos ruído também parece resultar em linhas de pauta mais degradadas.
## Contents

1 Introduction ........................................... 1
   1.1 OMR Systems ....................................... 2
   1.2 Motivation and Objectives ......................... 2
   1.3 Project Contribution ............................... 4
   1.4 Structure of the Report ............................ 4

2 State of the art ....................................... 5
   2.1 Global Thresholding .................................. 6
      2.1.1 Methods based on histogram shape .............. 6
      2.1.2 Clustering Methods ............................. 9
      2.1.3 Entropy Based Thresholding ...................... 11
      2.1.4 Thresholding Based on Attribute Similarity .... 13
      2.1.5 Spacial Thresholding Method ................... 16
   2.2 Adaptive Thresholding .............................. 17
      2.2.1 Local Variance Methods ......................... 17
      2.2.2 Local Contrast .................................. 17
      2.2.3 Center-Surround Schemes ......................... 18
      2.2.4 Surface-Fitting Thresholding .................... 19
      2.2.5 Edge Filling Threshold ........................ 20

3 BLIST ................................................ 23
   3.1 Robust estimation of staffline thickness and spacing in the gray-scale domain .... 23
   3.2 Content aware music score binarization ............. 24
      3.2.1 Using other reference lengths to guide the binarization ...... 25
      3.2.2 Adaptive content aware music score binarization .......... 26
   3.3 Code ................................................. 27

4 Methodology, Evaluation Metrics and Results .......... 31
   4.1 Methodology .......................................... 31
      4.1.1 Dataset ......................................... 32
      4.1.2 Output ........................................... 32
   4.2 Evaluation Metrics .................................. 32
      4.2.1 Results with Staff Finder Algorithms ............. 33
      4.2.2 Difference from Reference Threshold (DRT) ........... 33
      4.2.3 Misclassification Error .......................... 36
      4.2.4 Subjective Evaluation ........................... 37
## CONTENTS

4.3 Experimental Results and Analysis ........................................... 39  
4.3.1 Parameterization ............................................................. 39  
4.3.2 Results and Analysis ......................................................... 39  
4.4 Reproducible Research ......................................................... 42  

5 Conclusion and Future Work .................................................. 45  
5.1 Objectives Completed .......................................................... 45  
5.2 Future Work ................................................................. 45  

References .............................................................................. 47
## List of Figures

1.1 OMRSys Architecture (taken from [?]) ........................................... 2  
1.2 A sample music score. ................................................................. 3  
1.3 Otsu’s binarization of Figure 1.2. .................................................. 3  
1.4 A sample music score. ................................................................. 3  
1.5 Chen’s binarization of Figure 1.4. .................................................. 3  

2.1 Histogram of Figure 1.2 ................................................................. 6  
2.2 Representation of Sezgin and Sankur’s [MB04] classification. ............... 7  
2.3 Convex Hull and the histogram of an image. ...................................... 8  
2.4 Using the right and left peaks to determine $g_{new}$ in Yanni’s Method [YH94] 11  
2.5 $N_s(t)$ for an arbitrary $s$ ........................................................... 14  
2.6 A sample music score. ................................................................. 17  
2.7 Binarization with a global method. ............................................... 17  
2.8 5x5 window with large contrast ..................................................... 18  
2.9 5x5 window with close contrast ................................................... 18  
2.10 9x9 window used in Palumbo’s method. ........................................ 19  
2.11 Yanowitz-Bruckstein Method being applied to an image (adapted from [YB]). 20  
2.12 "Ghost" Object (adapted from [YB]). ........................................... 21  

3.1 Illustration of the estimation of the reference value line_thickness+spacing using a single column. In practice, sums of consecutive runs are accumulated over the whole image. ......................................................... 24  
3.2 Original music score (detail). ......................................................... 25  
3.3 Histogram of the sum of two consecutive runs. ................................... 25  
3.4 Binarization of Figure 3.2 with reference length: line_thickness+spacing. 26  
3.5 Binarization of Figure 3.2 with reference length: line_2thickness+spacing. 26  
3.6 Adaptive binarization of Figure 3.2 with reference length: line_thickness+spacing. 27  
3.7 Adaptive binarization of Figure 3.2 with reference length: line_2thickness+spacing. 27  

4.1 Sample image 12 from dataset. ..................................................... 33  
4.2 Sample image 53 from dataset. ..................................................... 34  
4.3 Binarization of image 02 from dataset with $t=210$ (detail) ................. 35  
4.4 Binarization of image 03 from dataset with $t=220$ (detail) ................. 35  
4.5 Binarization of image 47 from dataset with $t=110$ (detail) ................. 35  
4.6 Binarization of image 47 from dataset with $t=230$ (detail) ................. 35  
4.7 Image 63 from dataset (detail). ..................................................... 36  
4.8 Ground truth of Figure ?? ........................................................... 36  
4.9 Image 02 from dataset (detail). ..................................................... 37
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.10</td>
<td>Binarization of Figure 4.9</td>
<td>37</td>
</tr>
<tr>
<td>4.11</td>
<td>Image 01 from dataset (detail)</td>
<td>38</td>
</tr>
<tr>
<td>4.12</td>
<td>Binarization of Figure 4.11</td>
<td>38</td>
</tr>
<tr>
<td>4.13</td>
<td>Image 04 from dataset (detail)</td>
<td>38</td>
</tr>
<tr>
<td>4.14</td>
<td>Binarization of Figure 4.13</td>
<td>38</td>
</tr>
<tr>
<td>4.15</td>
<td>Binarization with Otsu method (detail)</td>
<td>41</td>
</tr>
<tr>
<td>4.16</td>
<td>Binarization with BLIST method (detail)</td>
<td>41</td>
</tr>
<tr>
<td>4.17</td>
<td>Binarization with Blist method (detail)</td>
<td>42</td>
</tr>
<tr>
<td>4.18</td>
<td>Binarization with Adaptive BLIST method (detail)</td>
<td>42</td>
</tr>
</tbody>
</table>
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCV2010</td>
<td>The Tenth Asian Conference on Computer Vision[acc10]</td>
</tr>
<tr>
<td>OMR</td>
<td>Optical Music Recognition</td>
</tr>
<tr>
<td>BLIST</td>
<td>Binarization based in Line Spacing and Thickness - The method proposed in this work</td>
</tr>
<tr>
<td>DRT</td>
<td>Difference from Reference Threshold</td>
</tr>
<tr>
<td>ME</td>
<td>Missclassification Error</td>
</tr>
<tr>
<td>MOPx</td>
<td>Missed Object Pixel</td>
</tr>
<tr>
<td>FOPx</td>
<td>False Object Pixel</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Physical documents deteriorate over time, which can lead to the loss of precious information. In order to preserve such documents, one of the most common processes is to digitalize them. For this task, many dedicated national and international programs and projects have been created, focusing on the preservation of huge volumes of documents. This digitalization of documents brings great advantages like easy duplication, distribution and digital processing.

One of the great problems with this kind of approach, however, is the recognition of the characters on a digitalized document. Converting a printed or handwritten document to a digital format, the computer usually only saves pixel information, without interpreting any meaning from the writing. As an example, it is impossible for a computer to search for a word in a document in such conditions. The Optical Character Recognition (OCR) domain has the ambition of solving this problem, using image processing, pattern recognition, artificial intelligence, etc.

Like most problems, solutions vary according to the subject in study. It is easy to understand that character recognition in a text document, which has letters and numbers as the only components, is fairly different from musical characters and other annotations in a music score which has staff lines. In order to transform the printed and handwritten music scores into a machine-readable symbolic format like MusicXML, facilitating operations such as search, retrieval and analysis, an Optical Music Recognition (OMR) system is needed [BB92].

If, on one hand, the processing of computer generated (or synthesized) music scores has good success rates in commercial software like capella-scan [Cro10] the same is not true about manuscript scores. The different notations used by each author, the variations on size and intensity which are characteristic of hand-writting and the line irregularity make this task a lot harder.
1.1 OMR Systems

The OMRSys is an FCT project at INESCPorto and was devised with the objective of creating a database of manuscript scores (primarily of Portuguese composers) integrated in an online system, allowing the recognition of such scores with acceptable success rates [AC08].

Several preprocessing techniques like noise removal, blurring, deskewing, binarization, among others, are employed to improve the efficiency of the recognition process. After these operations, the OMR system can be divided in three main modules:

- recognition of musical symbols;
- reconstruction of the musical information to build a logical description of musical notation;
- construction of a musical notation model for its representation as a symbolic description of the musical sheet.

The first module is typically divided into three stages: staff lines detection and removal to obtain an image containing only the musical symbols; symbols primitives segmentation and recognition.

![OMRSys Architecture](taken from [?])

This work is part of the pre-processing step, even before the Music Symbol Recognition process begins. The gray-scaling and binarization are usually the first steps in OMR, transforming the digitalized image in the correct input for the next steps as seen in Figure 1.1.

1.2 Motivation and Objectives

In digital image processing, as in all signal processing, several different image techniques are used to pre-process the input, making it ready for the recognition steps. In this particular case, the input for the staff finder algorithms are binarized images, containing either completely white or completely black pixels, with values 1 or 0, respectively. This means the digitalized image must be analyzed, in order to determine what is useful (the objects, being the music symbols and staves) and what is not (the background, noise).

To automatize this binarization process (which would be tedious if done by hand) many algorithms have been proposed in the past, with the success rates depending on the problem at hand. Even so, the binarization of the music score seldom justifies significant
attention from researchers who invariably use a standard binarization procedure, such as the Otsu’s method [Ots79]. Analyzing several samples, it is possible to infer that the binarization results with such methods are not perfect, often resulting in information loss. This problem (as shown in Figure 1.3) can be due to the non-uniform lighting resulting from the digitalization of music scores contained in books, for example, which end up having a darker area in the book’s spine as shown in Figures 1.2 and 1.3.

![Figure 1.2: A sample music score.](image1.png) ![Figure 1.3: Otsu's binarization of Figure 1.2.](image2.png)

Other times, binarization methods cause loss in the staves connection, making it hard for staff finder algorithms to achieve good results, as seen in Figures 1.4 and ??

![Figure 1.4: A sample music score.](image3.png) ![Figure 1.5: Chen’s binarization of Figure 1.4.](image4.png)

In order to significantly improve the binarization process, conserving the information that is important to OMR, not only should one consider the raw pixel information as well as the a priori knowledge about the content of the documents whenever possible. Since the binarization procedure is usually the first step of the processing system, there is usually no information available about the image content to assist the binarization procedure. However, it is possible to extract content related information from the gray-scale music scores to guide the binarization procedure.

The objectives of this project are:
Introduction

1. to implement different state of the art binarization methods as preprocessing for the staff finder algorithms;

2. to create a new binarization method based on content knowledge of music scores;

3. to compare results on both the state of the art generic methods and the new one.

1.3 Project Contribution

As stated, the binarization is one of the first steps in music scores recognition, and has such, as a great impact on the final result of the whole recognition process. Developing binarization methods specific to the problem at hand has the potential of showing better performance than the generic methods on the subsequent operations.

This dissertation presents the following contributions for the preservation and the general access to musical and cultural heritage:

1. The introduction to the music analysis community of the Binarization based in Line Spacing and Thickness (BLIST) algorithm to binarize the music scores.

2. The creation of a database of real scores with its segmented references.

3. Analysis and study of different binarization methods applied to music scores, which has never been done.

The work related with the project where the dissertation is inserted waits for the result of the submission of the article:

Content aware music score binarization
Authors: Telmo Pinto, Ana Rebelo, Gilson Giraldi, Jaime S. Cardoso
Conference: Tenth Asian Conference on Computer Vision [acc10]
Year: 2010

In addition, the work was done following the methodology of Reproducible Research [Ven10] to allow future researches to be able to correctly reproduce the results presented here.

1.4 Structure of the Report

This report follows the introduction with State of the Art in Chapter 2 where state of the art binarization methods are listed and described to some extent. Chapter 3 describes the new method proposed (BLIST). In Chapter 4, methods chosen for implementation and some details about them are presented as well as the metrics used in the evaluation and the experimental results and analysis. The conclusion and future work, in Chapter 5, end this report.
Chapter 2

State of the art

In this chapter, the state of the art methods for image binarization are described. They are not presented relating to the problem of music scores at hand, rather referring to any generic image, as no previous study was done on binarization of music scores, neither was a method developed specifically for this purpose.

The fundament of all the automatic binarization methods is the same: to find a threshold intensity value that correctly divides the intensities of the image into two groups: those lower than the threshold get the value 0 and are considered object, the important part to be used in further steps; those higher than the threshold get the value 1 and are considered as part of the background. [SS02]

Even though all methods find a threshold, some find one globally, for each image, while others find a different threshold for each pixel or different area of the image individually. These methods are called local or adaptive thresholding.

Many of the thresholding methods (both global and adaptive) use the information of the gray-level histogram to binarize the image. This histograms show, for each possible intensity (usually from 1 to 256) how many of the image pixels have that same intensity as seen in Figure 2.1.

When an image has a great difference in intensity between its dark and light pixels, it is said to be bimodal, having two separate modes, one representing the background and the other the objects. When this is the case it is easy to determine the threshold for the image. It usually falls somewhere between the modes.

In many situations, however, this is not the case. The modes of an image can present juxtaposition and there can even be more than two distinguishable modes. In this cases, the threshold finding method's task is less obvious, so the method has to be robust to be able to work on all images [SS02]
Mehmet Sezgin e Bulent Sankur [MB04] have classified thresholding methods taxonomically in the following groups: methods based on histogram shape; clustering methods; entropy methods; methods based on attribute similarity; spacial methods; adaptive methods; This scheme (Figure 2.2) is adopted as the chapter structure.

2.1 Global Thresholding

2.1.1 Methods based on histogram shape

These methods choose the threshold value based on different properties in histogram shape.

2.1.1.1 Convex Hull Thresholding

Rosenfeld [RIT83] defined a thresholding method analyzing histogram concavities. The candidates for threshold are chosen among the points of biggest concavity between the convex hull and the histogram itself as seen in Figure 2.3.

Some object attributes like low busyness of the threshold image edges are used to select between competing concavities.

2.1.1.2 Peak-and-Valley Thresholding

By convolving the histogram function with a smoothing and differencing kernel, adjusting the smoothing aperture of the kernel and resorting to peak merging, Sezan [Sez85] is able to turn the histogram into a bimodal distribution. This convolution outputs the zero-crossings referring to the initial, maximum and final points in each mode ($s_i, m_i, e_i$,
respectively, for each i mode). The optimum threshold is found between the first mode’s final and second mode’s first zero-crossing:

\[
\text{Output} = [(s_i, m_i, e_i), i = 1, 2]
\]

\[
T_{\text{opt}} = \gamma e_1 + (1 - \gamma)s_2, 0 \leq \gamma \leq 1
\]

(2.1)

Tsai smooths the histogram with a Gaussian kernel (to eliminate fake peaks) searching for sharp curvature points afterwards [Tsa95a]. For the histogram \( h(g) \) of intensities \( g \), he defines a peak in a certain \( g \) when

\[
h(g) > h(g - 1) \text{ and } h(g) > h(g + 1)
\]

(2.2)

and a valley when

\[
h(g) < h(g - 1) \text{ and } h(g) < h(g + 1)
\]

(2.3)

or

\[
h(g) < h(g - 1) \text{ and } h(g) = 0
\]

(2.4)

The smoothing level is manipulated by varying the Gaussian kernel width and number of convolutions. A one dimensional Gaussian kernel \( \text{gauss}(g, \sigma) \) with standard deviation
$\sigma$ as a parameter is used on the convolution with the histogram $h(g)$:

$$\text{gauss}(g, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{g^2}{2\sigma^2}\right)$$ \hspace{1cm} (2.5)

The convolution $H(g, \sigma)$ between the Gaussian kernel and the histogram

$$H(g, \sigma) = \int_{-\infty}^{\infty} h(f) \cdot \text{gauss}(g - f, \sigma) df$$ \hspace{1cm} (2.6)

is applied iteratively until the final histogram has the desired number of peaks.

Another possible approach is Carlotto’s [Car], where the zero-crossings of the derivatives of the histogram (fingerprints) are analyzed to infer information from different scales-spaces about the shape of the histogram. When extracting peaks and valleys from an histogram, the first derivative’s fingerprints can be detected in a scale where the most significant variations on the intensity are still present. Afterwards, the same fingerprints can be found on increasingly smaller scales, until the precise location of the peak or valley can be determined.

2.1.1.3 Shape-Modeling Thresholding

Ramesh, Yoo and Sethi [RYS95] approach the binarization problem trying to correspond the histogram with a function where each frequency value of an intensity below the threshold is replaced by that same value and all the frequency values above the threshold are replaced by another value.
State of the art

This approximation is recursive, and uses the error function $E(t)$, choosing as the optimal threshold the value of $t$ that minimizes this error.

$$E(t) = \sum_{g=0}^{t} [g - m_1(t)]^2 + \sum_{g=t+1}^{g_{\max}} [g - m_2(t)]^2$$

for

$$m_1(t) = \frac{\sum_{g=0}^{t} g \cdot h(g)}{\sum_{g=0}^{t} h(g)}$$
$$m_2(t) = \frac{\sum_{g=t+1}^{g_{\max}} g \cdot h(g)}{\sum_{g=0}^{t} h(g)}$$

(2.8)

where $h(g)$ is the histogram for intensity $g$. This method has shown the best results in images where the histograms have normal distributions.

2.1.1.4 Luminance Thresholding

This method, defined by Khashman [KS07], finds the optimum threshold $t$ using the maximum intensity found and the mean intensity of the histogram:

$$t = |m - (g_{\max} - m)|$$

(2.9)

where $m$ and $g_{\max}$ are the mean and maximum intensities, respectively.

2.1.2 Clustering Methods

With the clustering methods, it is assumed that the objects manifest in the intensity histogram as clusters. For the grayscale images, clusters can be defined as intervals between valleys or local minimum points. A class is attributed to each interval and pixels which have their intensities belonging to a certain interval are part of the same class.

In binarization methods, the number of final clusters or classes is always two, in order to find a single value that separates them - the threshold. [MB04]

2.1.2.1 Clustering Thresholding

A popular binarization technique, based on the gray-level histogram of the image is Otsu’s method [Ots79], in which the threshold is determined in order to maximize de homogeneity of each cluster.

Knowing that clusters that are more homogeneous have less variance between their pixels, the optimal threshold will be the one that assures the lesser sum of the weighted variances of all the pixels in each cluster. The variance inside the clusters is:

$$\sigma_{\text{inside}}^2(t) = p_1(t)\sigma_1^2(t) + p_2(t)\sigma_2^2(t)$$

(2.10)
where \( p_i \) is the probability and \( \sigma^2_i \) is the variance of cluster \( i \) with \( i=1 \) for intensities below chosen threshold and \( i=2 \) for intensities above.

The method consists of the application of this formula to each possible threshold, that is, to each different intensity, and using as optimal threshold the one that assures the lesser \( \sigma^2_{\text{ inside}} \).

Otsu proposes a simpler method to solve the same problem, based on the knowledge that minimizing the variance inside clusters is the same as maximizing the variance between them:

\[
\sigma^2_{\text{between}}(t) = \sigma^2 - \sigma^2_{\text{inside}} = \sum_{i=1,2} p_i(t) \cdot \mu_i(t) \cdot (\mu_i(t) - \mu(t))^2
\]  

(2.11)

where \( \mu_i \) is the mean of the cluster. This way, the method would find the threshold that maximizes equation 2.11, instead of the threshold that minimizes equation 2.10.

### 2.1.2.2 Iterative Thresholding

Ridler and Calvard [RC78] propose an iterative method that produces a threshold value from a series of background samples, each one closer to the objects, in order to exclude most of the background noise without being so close as to include the object itself.

In each iteration step, a new threshold is found, based on the mean of the background and object class means generated in the last iteration. When the difference between thresholds of adjacent steps is small enough, the iterations stop, returning the last threshold found.

Yanni and Horne [YH94] proposed an alternative method, in which the intensities \( g_{\text{max}} \) and \( g_{\text{min}} \) are determined. After this, a temporary middle point between \( g_{\text{max}} \) and \( g_{\text{min}} \) is defined as:

\[
g_{\text{med}} = \frac{g_{\text{max}} + g_{\text{min}}}{2}
\]  

(2.12)

Having determined \( g_{\text{med}} \), the local maximum intensities (or peaks) to the left and right of this point are found and they are used to determine a new middle point \( g_{\text{new}} \) as before, instead of the \( g_{\text{max}} \) and \( g_{\text{min}} \) (Figure 2.4). Threshold is determined using \( g_{\text{new}} \) based on the probability \( p(g) \):

\[
t = (g_{\text{max}} - g_{\text{min}}) \sum_{g=g_{\text{min}}}^{g_{\text{new}}} p(g)
\]  

(2.13)

### 2.1.2.3 Fuzzy Clustering Thresholding

An iterative binarization method is proposed by Jawahar, Biswas and Ray [JBR97], where the \( g \) points of an histogram \( h(g) \) are classified as either one of two fuzzy classes based
on the distance \( d \) of those points to the class means.

\[
\begin{align*}
    m_i(t) &= \frac{\sum_{g=0}^{g_{\text{max}}} g h(g) \mu_i(g)^\tau}{\sum_{g=0}^{g_{\text{max}}} h(g) \mu_i(g)^\tau}, \quad i = 1, 2 \\
    \mu_1(g) &= \frac{1}{1 + \frac{d(g, m_1)/d(g, m_2)}{\tau-1}} \\
    \mu_2(g) &= 1 - \mu_1(g)
\end{align*}
\]  

(2.14)

where \( \tau \) is the fuzziness index.

In each iteration step, the mean values are recalculated and each pixel is reassigned to the group closer to its position.

### 2.1.3 Entropy Based Thresholding

On a different approach to thresholding, some entropy methods have been proposed. Some of them interpret the maximization of the entropy on the binarized image as the maximization of the information that got through the process. Others, consider minimizing the cross-entropy between the original gray-scale and resulting binarization, as this also means the difference between the original and resulting images is minimized, preserving the maximum amount of information.

#### 2.1.3.1 Entropic Thresholding

Kapur [KSW85] described a threshold method based on the maximization of the entropy of the binarized image. The optimal threshold would be the one that assured the greater sum of the entropies of both classes:

\[
t_{opt} = \max[E_1(t) + E_2(t)]
\]  

(2.15)
The entropy functions for each class would be:

\[ E_1(t) = - \sum_{g=0}^{t} \frac{p(g)}{P(t)} \log_2 \frac{p(g)}{P(t)} \]  

(2.16)

for the objects and

\[ E_2(t) = - \sum_{g=t+1}^{g_{\text{max}}} \frac{g_{\text{max}}}{P(t)} \log_2 \frac{p(g)}{1 - P(t)} \]  

(2.17)

for the background, where \( p(g) \) is the probability of histogram point \( g \) and \( P(t) \) is the sum of all the probabilities up to threshold \( t \).

Taking this concept further, Sahoo, Wilkins and Yeager [SWY97] defined a correlation method as the weighted average of three different entropy sums, each generated with a different parameterization. The entropy function used is Renyi’s entropy, defined as:

\[ RE_1^\alpha = \frac{1}{1 - \alpha} \ln \left( \sum_{g=0}^{t} \left[ \frac{p(g)}{P(t)} \right]^\alpha \right) \]  

(2.18)

for the objects and

\[ RE_2^\alpha = \frac{1}{1 - \alpha} \ln \left( \sum_{g=t+1}^{g_{\text{max}}} \left[ \frac{p(g)}{1 - P(t)} \right]^\alpha \right) \]  

(2.19)

for the background where \( p(g) \) and \( P(t) \) are the same as above. This method finds three different thresholds, \( t_1, t_2 \) and \( t_3 \) for \( a < \alpha < 1, \alpha = 1 \) and \( \alpha > 1 \), respectively. The optimum threshold is determined as the weighted average of this three values.

Still based on this class entropy maximization concept, Albuquerque, Esquef and Mello [dAEM04] use another entropy function, Tsallis entropy:

\[ TE_1^q = \frac{1 - \sum_{g=0}^{t} [p(g)/P(t)]^q}{q - 1} \]  

(2.20)

for the objects and

\[ TE_2^q = \frac{1 - \sum_{g=t+1}^{g_{\text{max}}} [p(g)/(1 - P(t))]^q}{q - 1} \]  

(2.21)

for the background, where the real number \( q \) is the entropy index which should depend on the problem. Arguing that that a pseudo-additive entropy sum function works better with the presence of non-additive information in some classes of images that the purely additive entropy sum function presented by Kapur they presented the following:

\[ TE^q(t) = TE_1^q(t) + TE_2^q(t) + (1 - q).TE_1^q(t).TE_2^q(t) \]  

(2.22)

which should lead to the optimum threshold that maximizes the function.
2.1.3.2 Cross-Entropic Threshold

Li and Tam [LT98] defined the means of each class of the binarized image’s histogram as:

\[
\begin{align*}
    m_1(t) &= \frac{\sum_{g=0}^{t} g h(g)}{\sum_{g=0}^{t} h(g)} \\
    m_2(t) &= \frac{\sum_{g=t+1}^{t_{\text{max}}} g h(g)}{\sum_{g=t+1}^{t_{\text{max}}} h(g)}
\end{align*}
\]  

and minimize the following function, finding the optimum threshold:

\[
\eta(t) = - \sum_{g=0}^{t} h(g) \log[m_1(t)] - \sum_{g=t+1}^{t_{\text{max}}} h(g) \log[m_2(t)]
\]  

2.1.4 Thresholding Based on Attribute Similarity

These methods, as the name suggests, calculate the threshold value based on some similarity measure between the gray-scale and the binarized image. This attributes can be edge matching, shape compactness, gray-level moments, connectivity, texture, stability of objects, similarity as an imprecision measure, similarity of the cumulative probabilistic distributions or information revealed as part of the binarization.

2.1.4.1 Moment Preserving Thresholding

Tsai [Tsa95b] considers the gray-level as a blurred version of the optimal binarization. Defining moment m(k) for the gray-scale image as:

\[
m(k) = \sum_{g=0}^{t_{\text{max}}} p(g) g^k
\]  

where \(p(g)\) is the probability of histogram point \(g\) and for the binarization:

\[
b(k) = p_1.m_1(k)^k + p_2.m_2(k)^k
\]  

where \(p_1\) and \(p_2\) are the probabilities and \(m_1\) and \(m_2\) the separate moments of each one of the classes (object and background). The method finds a threshold that guarantees moment preserving, that is, the first three moments (for \(k=1, 2, 3\)) of the gray-scale are equal to the corresponding moments of the binarization.

2.1.4.2 Fuzzy Similarity Thresholding

Huang and Wang [HW95], trying to reflect the distance from the gray-scale image to its binarization, represent each pixel as a pair of attributes:

\[
F = \{I(x,y), \mu[I(x,y)]\}, 0 \leq \mu[I(x,y)] \leq 1
\]
where \( I(x,y) \) is the position of a pixel in a two-dimensional array that is the image and \( \mu[I(x,y)] \) represents its fuzzy membership to the object class.

Given the fuzzy memberships of all the pixels in an image, one can generate a fuzziness index \( FI \) for the whole image, using Shannon’s entropy [Sha48] in a two-dimensional space, simplifying it with the histogram \( h(g) \):

\[
FI(g) = \frac{1}{\frac{g_{\max}}{\ln(2)}} \sum_{g=0}^{g_{\max}} S(\mu(g))h(g)
\]  \hspace{1cm} (2.28)

The optimum threshold will be the one that minimizes this fuzziness index.

### 2.1.4.3 Topological Stable-State Thresholding

Pikaz and Averbunch [PA96] obtain the stable threshold when the objects reach their correct size.

Determining a function \( N_s(t) \) that represents the number of objects with at least \( s \) pixels, generated from the binarization with threshold \( t \), the optimum threshold is defined as being the mean value of the widest plateau found in \( N_s(t) \), that is, the most stable value for \( s \), as can be seen in Figure 2.5

![Figure 2.5: \( N_s(t) \) for an arbitrary \( s \)](image)

### 2.1.4.4 Maximum Information Thresholding

Leung and Lam [LL98] define the problem as a variation of uncertainty.

Let \( N \) be the number of pixels of the image and \( \alpha \) the probability of a pixel belonging to the object class, one can define \( \alpha N \) the object pixels and \( (1 - \alpha)N \) the background
pixels. On the histogram $h(g)$:

$$h(g) = \alpha h_1(g) + (1 - \alpha) h_2(g) \quad (2.29)$$

By application of the Shannon’s entropy function [Sha48]:

$$E(X) = -\alpha \log(\alpha) - (1 - \alpha) \log(1 - \alpha) = F(\alpha, 1 - \alpha) \quad (2.30)$$

where $E(X)$ is the uncertainty about which classe pixel $X$ belongs to. For a certain $g$ of the gray-scale image observed after the process, the corresponding $X$ has a conditional probability $P$ of belonging in the object class:

$$P(X \in 1 | g) = \frac{\alpha h_1(g)}{h(g)} \quad (2.31)$$

and a conditional probability $Q$ of belonging in the background class:

$$Q(X \in 1 | g) = \frac{(1 - \alpha) h_1(g)}{h(g)} \quad (2.32)$$

The uncertainty to which class $X$ belong to, given a certain $g$ is:

$$E(X | g) = F\{P(X \in h_1 | g), Q(X \in h_1 | g)\} \quad (2.33)$$

that is:

$$F\{\frac{\alpha h_1(g)}{h(g)}, \frac{(1 - \alpha) h_1(g)}{h(g)}\} \quad (2.34)$$

The weighted sum of equation 2.34 describes the average uncertainty for the whole image:

$$E(X | g_{max}) = \sum_{g=0}^{g_{max}} h(g). F\{\frac{\alpha h_1(g)}{h(g)}, \frac{(1 - \alpha) h_1(g)}{h(g)}\} \quad (2.35)$$

With some mathematical manipulation this function can also be written as:

$$E(X | g_{max}) = E(X) + \alpha E(h_1) + (1 - \alpha) E(h_2) - E(h) \quad (2.36)$$

Now it is possible to define the reduction on uncertain pixel classification after the observation of the gray-scale image:

$$\text{initial uncertainty} - \text{residual uncertainty} = E(x) - E(X | g_{max}) = E(h) - \alpha E(h_1) + (1 - \alpha) - E(h_2) \quad (2.37)$$
2.1.4.5 Enhancement of Fuzzy Compactness Thresholding

Making use of fuzzy geometry concepts defined by Rosenfeld [RIT83] such as area and perimeter of a fuzzy set given by the fuzzy membership of a pixel to a class (object or background). Thus, the fuzzy set is defined as function of the coordinates $x$ and $y$ of the image pixel:

\[
\text{Area}[\mu(x,y)] = \sum_{x=0}^{x_{\max}} \sum_{y=0}^{y_{\max}} \mu(x,y) \quad (2.38)
\]

\[
\text{Perim}[\mu(x,y)] = \sum_{x=0}^{x_{\max}} \sum_{y=0}^{y_{\max}-1} |\mu(x,y) - \mu(x,y+1)| + \sum_{x=0}^{x_{\max}-1} \sum_{y=0}^{y_{\max}} |\mu(x,y) - \mu(x+1,y)| \quad (2.39)
\]

S. K. Pal [PR88] defined his thresholding method by determining the optimum threshold that maximizes the compactness of the objects:

\[
\text{Comp}[\mu(x,y)] = \frac{\text{Area}[\mu(x,y)]}{\text{Perim}[\mu(x,y)]^2} \quad (2.40)
\]

2.1.5 Spacial Thresholding Method

These algorithms use information of the neighbor pixels, besides intensity histogram information, to determine the threshold.

N. R. Pal [PP89], for example, notes that although two images can have similar histograms, their $n$-th order entropies can be different, so it is important to consider the cooccurrence probability of the gray-values in neighbor pixels. He then proposes two different methods, one in which the image is forced to have as many between class transitions as possible and another one in which the opposite is true.

Chang, Chen, Wang and Althouse [CCWA94] consider the cooccurrence between pixels of the gray-scale image and its binarization, classifying transitions between pixels in four distinct groups, according to their behavior relating a possible threshold: background-background, object-background, object-object, background-object. The optimum threshold is the one that minimizes the divergence of the images.

Abutaleb [Abu89] starts with the concept of a 2-D histogram $h(g,f)$, where $g$ is the gray-level of a pixel and $f$ the average gray-level among the neighbors of that pixel. Using a second order entropy function it is possible to determine the optimum threshold.

Xiao [XCZ08] takes into account not only the gray-levels of each pixel, but also their similarity with neighbor pixels, building a two-dimensional histogram with what he calls gray-level spacial correlation. He then uses the maximization of Shannon’s entropy [Sha48] on this histogram to determine the threshold.
2.2 Adaptive Thresholding

Many times, in digitalizations of real documents, the light levels vary through the page. A page from a book for example, tends to become darker near the spine of the book, as seen in Figure 2.6.

The application of global thresholding methods, although resulting in acceptable binarizations in many cases, can be proven ineffective in the fore-mentioned cases where the illumination is not homogenous similarly to what happened in Figure 2.7.

![Figure 2.6: A sample music score.](image)

The goal of local or adaptive thresholding methods is to solve such problems, by finding an independent threshold for different segments, windows, sometimes even pixels of the image, instead of a single value for the whole document.

2.2.1 Local Variance Methods

Niblack [Nib86] adapts his threshold for a window, according to the mean \( m(x,y) \) and standard deviation \( \sigma(x,y) \) of the pixels (the chosen window should produce acceptable results, being small enough to preserve detail and big enough to suppress noise). For each window the threshold is defined as:

\[
T(x,y) = m(x,y) + k\sigma(x,y)
\]  

(2.41)

where \( k \) is a value that indicates how much of the margins is part of the object. Both this \( k \) and window size should be chosen experimentally, according to the problem.

2.2.2 Local Contrast

Contrast can also be used to determine threshold locally. White and Rohrer [WR83] developed a binarization method for documents where characters are to be identified, in which the intensity of each pixel is compared to the mean intensity of the neighbor pixels.
State of the art

Once again, the size of the neighborhood considered is crucial, this time because it should be the approximate size of a typical character. If the pixel is significantly darker than the neighborhood mean, it is classified as object, otherwise it is considered background.

In Bernsen’s method [Ber86] the pixels with the highest and lowest intensity of each window are considered. The mean of these two values represents the threshold to be used for that area (Figure 2.8), unless the contrast between these two pixels is not big enough (Figure 2.9), in which case the whole window is considered to be the same class. This last process depends on another threshold value for contrast, defined for the whole image.

![Figure 2.8: 5x5 window with large contrast](image)

![Figure 2.9: 5x5 window with close contrast](image)

2.2.3 Center-Surround Schemes

Palumbo, Swaminathan and Srihari [PSS86] use five threshold values \( t_1, t_2, \ldots, t_5 \) for the whole image, defined before the method is applied, even though the binarization is really a local process.

Considering a 9x9 window, the 3x3 center square \( C_{center} \) is responsible for object capture. Defining another group of pixels in the same window \( C_{diagonal} \) composed by the
four 3x3 squares diagonal to $C_{\text{center}}$, responsible for background capture, it is said that the $C_{\text{center}}$ pixels are the most significant and the $C_{\text{diagonal}}$ the least significant (with less impact on the final result). The remaining pixels in the 9x9 window are not used. This window scheme can be seen in Figure 2.10.

![Figure 2.10: 9x9 window used in Palumbo’s method.](image)

All the points in the C sets defined before which are lower than $t_1$ are considered as object. The pixels in $C_{\text{diagonal}}$ that are lower that $t_2$ are also considered object if:

$$M_{\text{diagonal}2}.t_3 + t_5 > M_{\text{center}}.t_4$$  \hspace{1cm} (2.42)

Where $M_{\text{diagonal}2}$ is the mean of the pixels in $C_{\text{diagonal}}$ that are lower that $t_2$ and $M_{\text{center}}$ is the mean of the values in $C_{\text{center}}$.

### 2.2.4 Surface-Fitting Thresholding

Yanowitz and Bruckstein [YB] defined a method that combines gray-level and edge information to create a threshold surface to apply to the image.

Initially, the image is smoothed substituting each pixel for the mean intensity of a small neighborhood. This process makes variations in intensity more moderate and less error prone, making the next steps easier.

Next, the gradient magnitude is derived from the image, from which the local maximum (or edges of the objects) are determined. These local maximum are good candidates for local threshold values and are used in the following step, being interpolated, determining the threshold surface.

In Figure 2.11 the gray-levels for a y-slice of a 3D representation of the image can be seen as well as the gradient magnitude and corresponding interpolation points.
Applying the threshold surface to the image, it is possible to binarize it, having an independent threshold for each pixel.

Additionally, due to the possibility of the occurrence of "ghost objects", or false objects generated from large noise stains in the image, a validation system is used. Starting with the binarized image, the process is to search for connected-components. For each one of these found, the edges are found in the gradient map generated before. If, for a certain connected-component, the mean of the values in it’s edges do not exceed a certain value, they are eliminated.

2.2.5 Edge Filling Threshold

Chen [CsSHsX08], described a double-threshold image binarization method which uses Canny edge detection [Can86].
State of the art

Figure 2.12: "Ghost" Object (adapted from [YB]).

Starting with the original image, this method starts by applying the edge detection. Next, using the 8-neighborhood of each edge pixel, the seed point for each pixel - the lowest intensity in the neighborhood of said pixel - and the low and high intensities - the ones whose gradient magnitudes are larger than the high threshold of the Canny edge detector - are determined.

Using this low and high intensities of all the edge pixels, two thresholds are defined: one as the average of the high intensities, the other as the average of the low intensities. The image is binarized the first time using the high threshold, and then uses the seed points to fill it’s partitions. A second binarization using the low threshold should then be combined with the result from the seed filling step, determining which of the filled areas are objects.

While this method uses two global thresholds for the whole image, it uses them to fill the partitions found with edge detection, bringing it closer to the adaptive threshold methods.
State of the art
In order to introduce the new method developed in this work, some theoretical concepts must be described beforehand in the next sections.

The following text was adapted from the article produced for the ACCV2010 [acc10] conference.

3.1 Robust estimation of staffline thickness and spacing in the gray-scale domain

The conventional estimation of the staffline thickness and spacing assumes the run-length encoding (RLE) of each column of the binary music score. In this representation, the most common black-run is likely to represent the staffline thickness and the most common white-run is likely to represent the staffline spacing. Even in music scores with different staff sizes, there will be prominent peaks at the most frequent thickness and spacing. These estimates are also immune to severe rotation of the image [KI92, Fuj04].

In [CR10] the authors suggest to estimate directly the sum of the staffline thickness and spacing, hereafter termed line_thickness+spacing, since this can be robustly estimated by finding the most common sum of two consecutive vertical runs (either black run followed by white run or the reverse). The process is illustrated in Figure 3.1.

Moreover, instead of computing the most frequent peak in the histogram of the runs for a binarized image (binarized with a state-of-the-art binarization method), the authors propose to compute the histogram of the runs for ‘every’ possible binary image, by accumulating the runs’ frequency when varying the binarization threshold from the lowest to the highest possible values. This is illustrated in Figures 3.2 and 3.3, with the histogram of the sum of two consecutive runs. In this example, the estimation for line_thickness+.spacing results in 24.
This procedure of computing the reference length \texttt{line_thickness+spacing} without assuming any binarization threshold, allows the extraction of important information directly from the gray-scale image. We propose now to use this information to guide the binarization procedure.

3.2 Content aware music score binarization

As stated in the introduction, an OMR system typically encompasses, in one of its first steps, the detection of the stafflines to facilitate the subsequent operations. An image binarization method that maximizes the ‘presence’ of the lines in the binarized image may contribute significantly to the improvement of the following operations.

A binarization method designed to maximize the number of the pairs of consecutive runs summing \texttt{line_thickness+spacing} (the peak computed over the gray-level image) will likely maximize the quality of the binarized lines. However, the direct maximization of the count of pairs of consecutive runs summing \texttt{line_thickness+spacing} could lead to a threshold value producing many, ‘noisy’, runs, and as a side effect, many runs at \texttt{line_thickness+spacing}. The use of relative histograms is also prone to problems since now one may end up choosing a threshold with a very low absolute count of runs in \texttt{line_thickness+spacing} but that, by chance, is the highest relative count.

Therefore, the candidate thresholds are restricted to those producing a histogram of runs with the mode at \texttt{line_thickness+spacing}. If no threshold is found with this condition (note that even if the integration over all thresholds does have a mode at \texttt{line_thickness+spacing}, it is possible that no individual threshold produces a histogram with mode at \texttt{line_thickness+spacing}), the minimum integer $i$ for which there are threshold values with histogram mode at

![Figure 3.1: Illustration of the estimation of the reference value line_thickness+spacing using a single column. In practice, sums of consecutive runs are accumulated over the whole image.](image)
line\_thickness+spacing \pm i is considered. From the set of candidate thresholds, the proposed binarization method for music scores simply selects the threshold that maximizes the count of pairs of consecutive runs on the mode.

3.2.1 Using other reference lengths to guide the binarization

The same rationale used to motivate the estimation of the sum of pair of consecutive lengths, can be used to work with sets of three or more consecutive runs. However, two problems arise when proceeding that way: there is the underlying assumption that each staff have enough lines to give meaning to the consecutive runs and one starts getting less and less values to accumulate in the histogram, potentially leading to less accurate estimations.

A potentially interesting balance is estimating the sum of two times the line thickness plus the spacing, line\_2\text{thickness+spacing}, by working with the frequencies of triplets (black run, white run, black run). This only assumes that each staff has at least two lines, but does impact the number of accumulated values, roughly halving it. The proposed content aware binarization method does not suffer any adaptation, besides the change of the reference length, line\_thickness+spacing by line\_2\text{thickness+spacing}. Further on, in this report, the two options will be compared. In Figures 3.4 and 3.5 the results obtained with the proposed approach are illustrated, using the two aforementioned reference lengths. One can observe that the resulting stafflines have good quality, with minor differences between the two results.

Nevertheless, the original music score in this particular example is not correctly binarized with a global threshold. The digitalization of bound documents, such as books, either performed by flatbed scanners or digital cameras often yields images that exhibit a gradient-like distortion in the average colour in the region close to the book spine. In these
cases, adaptive methods can show better performance as mentioned before in Chapters 1 and 2.

### 3.2.2 Adaptive content aware music score binarization

Despite having been presented as a global thresholding method and having been applied it to the whole image, nothing prevents the application of the just developed ideas to a sampling window around a pixel $p$, effectively converting the proposed method to a local method.

As with other adaptive methods, the size of the sampling window is a key parameter. With this approach, the sampling window should be big enough to accumulate enough information (runs) to provide a proper solution. Since the typical distortions in this kind of documents are vertically oriented, the local threshold should be constant along a column of the image. Therefore it is suggested to compute a single threshold per column, using as window a vertical strip with height equal to the height of the image and width defined by the user.

One major barrier in the application of adaptive and local models is their high computational cost. Usually, these models require the estimation of several statistical variables for each pixel of the input image (in our case for each column of the image). Depending on the size of the sampling window, the computational cost can be very high. Traditional solutions to this problem have been interpolating techniques, in which the threshold value is computed on a set of sampled columns, and then, for the rest of the columns, the threshold value is calculated by interpolating on the sampled columns.

In Figures 3.6 and 3.7 the results obtained with the proposed approach are illustrated, using a window width and step size of 2% of the width of the image, and cubic polynomial interpolation.

In this example, the adaptive method using the \texttt{line\_thickness}+\texttt{spacing} reference length provided the best results, with a better staffline definition.
3.3 Code

In this section a sample pseudo Matlab code implementation of BLIST is presented.

function BLIST(img, method)

    [nrows ncols] = size(img);
    medvalue = median(img(:));
    minValue = min(img(:));

    % the maximum value of the sum of two or three consecutive runs is nrows
    acumHist = zeros(nrows,1);
    % histogram for each threshold
    indHists = zeros(nrows,256);

    for th=minValue+1:medvalue
        bw = (img>th);
        for col=1:ncols
            data = bw(:,col);
            % run function for RLE
            data = rle(data);
            values1=data(2);
            values2=values1(2:end);
            values3=values1(3:end);

            % check what method is being used
            if method=='pairs'
                sumConsecutiveRuns=values1(1:end-1)+values2;
            else if method=='triplets'
                sumConsecutiveRuns=values1(1:end-2)+values2(1:end-1)
... +values3;

% only accept triplets (black, white, black)
sumConsecutiveRuns=sumConsecutiveRuns((bw(1,col)
... +1):2:end);
end

% fill the runs' histograms
for i=1:length(sumConsecutiveRuns)
    indHists(sumConsecutiveRuns(i), th) =
        ... indHists(sumConsecutiveRuns(i), th) + 1;
    acumHist(sumConsecutiveRuns(i)) =
        ... acumHist(sumConsecutiveRuns(i)) + 1;
end
end

% find peak in the sum of the histograms
[unusedValue referenceLength] = max(acumHist)

% find the peaks for the histograms of all thresholds
[value unusedValue] = max(indHists, [], 1);
i=0;
idx=[];

% find the lowest threshold histogram that has the peak in the same
% index as the sum of all histograms
while isempty(idx)
    % don't let the index go beyond nrows
    ref = min(nrows, referenceLength+i);
    idx = find(value==indHists(ref,:));
    if ~isempty(idx)
        break;
    end
    % don't let the index drop below 1
    ref = max(1, referenceLength-i);
    idx = find(value==indHists(ref,:));
% if no threshold was found, try the same for the peak+i, then for peak-i
    i = i + 1;
end
BLIST

[unusedValue bestThr] = max(value(idx))
threshold = idx(bestThr);
return
BLIST
Chapter 4

Methodology, Evaluation Metrics and Results

This chapter describes method specification details and the evaluation methods to be used.

4.1 Methodology

Because all the methods would undergo similar tests, the inputs and outputs should all be the same, allowing the creation of scripts that would run the implementations at once, producing standardized results for comparison.

Unfortunately, due to time constraints on this project, the implementation of all the methods presented in the state of the art would be unrealistic. Therefore, the need for a choice among methods proven to be a necessity.

The methods were chosen in order to encompass different approaches to binarization, while still remaining relevant for the problem at hand.

Otsu’s method [Ots79], being one of the most famous and referenced binarizations due to its simplicity and efficiency [TJ95, TT95] was a natural choice as a clustering method. The Matlab function `graythresh` was used as the implementation of this algorithm.

Entropic methods, shown to be popular and efficient [MB04], were also represented by Kapur’s method, which presents the original idea of maximization of entropy in binarized images as a maximization of the information that got through the process. Sahoo’s Correlation method [SWY97] shows work upon the original idea, having a different entropy function dependent of parameterization and applying this function in three different parameterizations to the binarization, correlating the results in a weighted average. The entropy method by Albuquerque [dAEM04] not only used Tsallis entropy function but
also makes use of a pseudo-additive function to add the entropies of different classes, which, the authors argue, presents better results.

As for methods based on attribute similarity, two methods were chosen: Tsai’s thresholding [Tsa95b] that searches for the binarization that preserves moment and Huang’s fuzzy similarity thresholding [HW95] that studies the fuzzy membership of the pixels to each class.

Khashman’s luminance method [KS07] was developed specially for paper-based documents. It was chosen for it’s simplicity to implement and alternative approach, using a fast formula that considers the highest and mean intensity values of each image.

As for adaptive methods, Niblack’s method [Nib86] that calculates an independent threshold for each window instead of the whole picture was chosen. Bernsen’s method [Ber86], not only determines threshold pixel by pixel but it also uses contrast information.

Yanowitz [YB] binarization is a fairly more complicated process, determining a threshold surface for all the pixels at the same time, based on edge detection.

Chen’s proposed method [CsSHsX08] has a "hybrid approach" which uses only two global thresholds for each image, but uses them to fill areas found with the Canny edge detector [Can86], and which can produce interesting results.

Both a global and a local version of BLIST were implemented. Because of the good results obtained by Otsu’s method in the global method analysis (as will be seen in section 4.3), an adaptive version of Otsu’s method was also implemented, using the same reasoning as with BLIST, described in the last chapter.

4.1.1 Dataset

The dataset was composed of 65 digitalized handwritten music scores from six different authors. Before any method was applied, all the images were reduced to their intensity values (gray-scale) with no further pre-processing.

The images were not altered to produce or correct deformations, having only those typically generated by the digitalization process and the scores’ natural degradation. Figures 4.1 and 4.2 are sample scores from the dataset.

4.1.2 Output

Each method should output the binarized score. Additionally, global methods also output the threshold value for each image.

4.2 Evaluation Metrics

In order to evaluate the methods proposed, a careful analysis must be prepared to assure that results represent true information.
Different evaluation methods and their purpose is described in this section.

4.2.1 Results with Staff Finder Algorithms

The most obvious of the evaluations proposed, consists on applying staff finding algorithms to the binarizations and measuring the resulting errors.

Following the input of the binary scores resulting from each thresholding method in the staff finding algorithms, the expected output would be the percentage of both missed lines and false positives.

As staff finding algorithms can work differently, using different approaches to reach the same end, two methods were tested: Stable Path algorithm [AC08, JSC08] and Dalitz algorithm [DDCF08].

The obvious advantage of this methodology is the value of the results. The staff finding process usually follows the pre-processing in OMR systems, so by assuring a binarization method can produce good staves one can have good indications of the value of a method within the image recognition process.

Even though this analysis can show which methods work better indirectly, by showing how the binarizations behave further on the OMRSys process, it may be hard to identify the reasons why such binarization produces those results. This shows a need for other evaluation methods, with more direct approaches.
4.2.2 Difference from Reference Threshold (DRT)

Another possible evaluation method would be to manually binarize the dataset, comparing these reference thresholds with the ones resulting from the methods.

The reference thresholds were supervised independently by five different people, who received the following instructions in order to find the threshold that produces the best results with the staff finder algorithms used in the next steps:

1. The highest priority feature in the binarizations should be the music staves;
2. If the music staves are assured, the next important thing is the noise;
3. If the noise does not distort or occlude the objects, then these objects should be as dark as possible.

This evaluation process is fast to apply and gives straightforward results: one can easily use the modulus of the difference of the reference and method output values and have an indicator of how many intensity levels the two are apart.

\[
DRT(score) = |ref\_value(score) - method\_threshold(score)|
\] (4.1)

This approach has some limitations, one being the fact that it can only be applied to global thresholding methods, as adaptive methods do not use a single thresholding value for the whole image.

Figure 4.2: Sample image 53 from dataset.
Additionally, as it will become clear in section 4.3, this method is not very accurate, as there are both cases of scores with a wide range of acceptable thresholds and cases of scores where the opposite is true. This means that in some scores if the method had a DRT of 100 intensity values it could still produce a good binarization whereas in some cases where the DRT is as low as 10, the resulting binarization can be dramatically useless. This can be seen in Figure 4.3 and Figure 4.4 where there is a small range of acceptable results and in Figure 4.5 and Figure 4.6 where there is a wide range of possible thresholds.

Even so, using an average of multiple supervised values (from different individuals)
can make the reference thresholds converge to a stable value, closer to the middle of the acceptable value range. Averaging the DRTs of all the images for each method also helps in this issue, diluting possible outliers.

### 4.2.3 Misclassification Error

Finally, another evaluation method can be defined by comparing binarized images to ground-truth binarizations, pixel-by-pixel.

The process to create ground-truths is to binarize images by hand, cleaning all the noise and background, making sure nothing more than the objects remains. This is illustrated in Figures 4.7 and 4.8.

![Figure 4.7: Image 63 from dataset (detail).](Figure 4.7)

![Figure 4.8: Ground truth of Figure ??](Figure 4.8)

After these reference binarizations are created, the goal is to compare each one of them to the result of each method for that same image, pixel-by-pixel, according to the following equation:

\[
ME = 1 - \frac{(B_{bin} \cap B_{gt}) + (F_{bin} \cap F_{gt})}{\#B_{bin} + \#F_{bin}}
\]  

In equation 4.2, \(B_{bin}\) and \(F_{bin}\) represent the background and foreground pixels of the binarization being tested, and \(B_{gt}\) and \(F_{gt}\) the background and foreground pixels in the reference ground-truth image, respectively. \# is the cardinality, or more precisely, the number of elements in a specific set.

The manual binarization process is not only tedious but also extremely time-consuming, as all the uncleared noise could wrongly reward the worst performance. To assure that the ground-truths are well generated, the process of doing so “by hand” has to take it’s time.

This ends up limiting the amount of scores which can have their ground truths generated. For this reason only ten scores were chosen from all over the dataset (assuring more than one author) to be turned into ground truths.
Methodology, Evaluation Metrics and Results

On the other hand, this method can be applied both to global and adaptive binarizations, and even allows comparison between both types of methods.

For the adaptive binarizations, however, as will be seen in section 4.3, the Misclassification Errors were all very similar. A further analysis was conducted, still based on ground truths of ten scores. Two new error rates are presented: the Missed Object Pixel rate in equation 4.3 and the False Object Pixel in equation 4.4, dealing with loss in object pixels and excess noise, respectively.

\[
MOP_x = \frac{\#F_{gt} - \#(F_{bin} \cap F_{gt})}{\#F_{gt}} \quad (4.3)
\]

\[
FOP_x = \frac{\#F_{bin} - \#(F_{bin} \cap F_{gt})}{\#F_{bin}} \quad (4.4)
\]

4.2.4 Subjective Evaluation

Although objective evaluation methods hold the factual value of numbers, a subjective analysis, obviously faulty due to the relative interpretation of each observer, can also help identifying problems with the binarization methods. This type of evaluation does not pretend to be perfect. Instead, it’s goal will be to try to help make some sense of some facts present in the objective evaluation.

In order to make this kind of analysis as standard and hold as much quantifiable value as possible, and objectify all that can be done so, three different errors were considered: Loss in staff connectivity; occlusion of information by noise; back-to-front interference.

Loss in staff connectivity happens when a binarized score has staves disconnected in places where they are not disconnected in the gray-level as can be seen in Figure 4.9 and Figure 4.10.

Figure 4.9: Image 02 from dataset (detail).  
Figure 4.10: Binarization of Figure 4.9
Occlusion happens when some information (other than staff lines) is covered by noise, making it impossible to distinguish what is there, as seen in Figure 4.11 and Figure 4.12.

Back-to-front interference happens when notes and characters from the back of the music score can be seen through the page after the digitalization like in Figure 4.13. A binarization, if done incorrectly, can also present such noise as objects, creating false information which should be accounted as an error like in Figure 4.14.

These errors have been arranged by importance level, by the negative impact they have on the following steps of the OMR. The disconnected staves is the worst of all errors, as music scores with missing staves are impossible to extract information from. Any missing staff on each group of five music staves will render the whole group useless, as the position of the notes relatively to the staves is what differentiates tones. The second worst error is the occlusion, as any character hidden behind noise will also be impossible to identify. Third, the presence of false information, created by back-to-front interference the least serious of this three errors.
The process for an observer would be to look at the binarizations resulting from the methods, identifying which of the three errors (if any) are present in each image.

### 4.3 Experimental Results and Analysis

#### 4.3.1 Parameterization

Entropy methods by Sahoo [SWY97] and Albuquerque both need additional inputs, parameterizing their entropy functions. Albuquerque requires the introduction of the value for \( q \) as seen in chapter 2. Sahoo requires two values of \( \alpha \), one being \( \alpha < 1 \) and the other \( \alpha > 1 \), in order to correlate different entropy functions for the same image.

As for Niblack’s [Nib86], the window size must be provided, plus a coefficient of the standard deviation, once again as seen in chapter 2.

Similarly, Bernsen’s [Ber86] needs window size (although this method runs pixel by pixel, it considers a neighborhood around said pixels). It also needs the input of a contrast threshold to determine if all the pixels in each neighborhood belong to the same class.

Chen’s method requires the input of many parameters: a standard deviation to be used in the Gaussian filter that smoothes the image, the number of seeds that have to be continuously connected, the ratio of noise removal and the radius of the window considered for connecting isolated points.

The Adaptive BLIST and Adaptive Otsu input the percentage of the total image size to be used as window width.

The input for each method was determined experimentally, by considering which parameters provided the best result for that method using the Misclassification Error.

- Tsallis [dAEM04]: \( q = 2 \)
- Sahoo [SWY97]: \( \alpha_1 = 0.4; \alpha_3 = 3 \)
- Niblack [Nib86]: window = 200; \( k = -1 \)
- Bernsen [Ber86]: window = 10; contrast = 20
- Chen [CsSHsX08]: \( \sigma = 3; n\text{SeedsConnected} = 15; \text{noiseRemoval} = 30; \text{radius} = 10 \)
- Ad BLIST, Ad Otsu: windowWidth = 2%

#### 4.3.2 Results and Analysis

The following text was adapted from the article produced for the ACCV2010 [acc10] conference.

The analysis will start with the results for the global methods presented in table 4.1.
Methodology, Evaluation Metrics and Results

<table>
<thead>
<tr>
<th>Otsu</th>
<th>BLIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loss Staff Connection</td>
<td>15.4</td>
</tr>
<tr>
<td>Occlusion</td>
<td>9.2</td>
</tr>
<tr>
<td>Back-to-Front Interference</td>
<td>10.8</td>
</tr>
</tbody>
</table>

Table 4.2: Subjective analysis of Otsu and BLIST methods, according to percentage of binarized music scores with loss in staff connection, occlusion and back-to-front interference, as explained in section 4.2.

Both versions of the BLIST method proposed (using pairs and triplets of consecutive runs), performed above average. Even so, the version that uses the pairs of runs instead of the triplets did better in the tests. This version will be considered on all the following comparisons.

Entropy based binarizations and Khashman’s algorithms got fairly similar results to each other. Huang and Tsai managed to top these results, with acceptable line detection rates and Misclassification Error. There are, however, two binarization techniques that get consistently better results than the others: Otsu’s Method and BLIST method. The only major difference is the higher missed staff detection rate for the Otsu’s algorithm.

To make a better sense of what is causing this differences between both algorithms, the subjective analysis (Table 4.2) of the results of the two methods can be useful. Once again, it can be said that the objective of this subjective analysis is not to take any definitive conclusions, instead it only tries to make observations a little more quantifiable than just plain text.
Methodology, Evaluation Metrics and Results

Figure 4.15: Binarization with Otsu method (detail).

Figure 4.16: Binarization with BLIST method (detail).

value that produces both the presence of perfectly connected staves and no occlusion of data with noise. Although staves can be correctly found in global thresholding procedures, adaptive methods can produce results with little or no loss of information.

The adaptive versions of the BLIST and Otsu’s methods were implemented as described previously with the window width used was a fixed percentage of the total image width and the interpolation of the threshold values obtained with a third degree polynomial regression.

As mentioned in section 4.2, when testing local thresholding methods, the DRT evaluation cannot be performed, as there is no single threshold value for each image. In addition, because the Misclassification Errors were all very similar, the MOPx and FOPx were found for each method. The results for the adaptive methods are presented in table ??.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ME: avg %</td>
<td>4.3</td>
<td>3.2</td>
<td>4.2</td>
<td>4.3</td>
<td>4.2</td>
<td>3.5</td>
</tr>
<tr>
<td>MOPx: avg %</td>
<td>24.6</td>
<td>22.5</td>
<td>15.6</td>
<td>22.5</td>
<td>21.7</td>
<td>12.4</td>
</tr>
<tr>
<td>FOPx: avg %</td>
<td>17.2</td>
<td>4.3</td>
<td>18.5</td>
<td>13.3</td>
<td>16.5</td>
<td>14.7</td>
</tr>
<tr>
<td>SP False: avg(std) %</td>
<td>1.3(3.0)</td>
<td>9.9(9.7)</td>
<td>2.1(3.6)</td>
<td>3.2(4.8)</td>
<td>2.7(3.9)</td>
<td>4.2(7.6)</td>
</tr>
<tr>
<td>SP Missed: avg(std) %</td>
<td>1.9(4.4)</td>
<td>33.0(32.8)</td>
<td>2.3(5.5)</td>
<td>14.2(23.2)</td>
<td>10.7(23.6)</td>
<td>7.9(13.8)</td>
</tr>
<tr>
<td>Dal False: avg(std) %</td>
<td>3.9(12.9)</td>
<td>3.4(36.6)</td>
<td>3.8(6.2)</td>
<td>3.1(5.1)</td>
<td>3.2(4.9)</td>
<td>3.5(6.1)</td>
</tr>
<tr>
<td>Dal Missed: avg(std) %</td>
<td>9.0(16.2)</td>
<td>17.3(27.0)</td>
<td>8.4(14.6)</td>
<td>10.2(14.1)</td>
<td>10.7(18.9)</td>
<td>7.7(10.1)</td>
</tr>
</tbody>
</table>

Table 4.3: Test results for various local thresholding methods, using different evaluations (in percentage): misclassification error, Missed Object Pixels, False Object Pixels, staff detection error rates for missed and false staves with Stable Path [AC08, JSC08] and Dalitz [DDCF08].

Ad BLIST and Yanowitz show the lowest MOPx, meaning these are the methods that find most of the correct pixels, which translates into lower missed staves rates. Even so, Ad BLIST also has a FOPx rate slightly higher than the other methods. This higher noise also translates into a slightly higher false staves rate with Dalitz method. Bernsen’s binarizations, although presenting the highest missed pixel rate, seem to perform well in the staff finding steps, having both the lowest missed and false staves rates.

Comparison between global and local thresholding suggests similar results between the best of each class of techniques. Even with the increase in computational cost for
adaptive methods, no significant improvement is show in the staff finding steps.

These local methods, however, may be proven useful with further testing. In some cases where both methods were capable of correctly finding the music staves, the noise produced with the global thresholding occludes some relevant information as seen in Figures 4.17 and 4.18. This may prove critical in the symbol recognition steps that follow the staffline detection.

![Figure 4.17: Binarization with Blist method (detail).](image1)

![Figure 4.18: Binarization with Adaptive BLIST method (detail).](image2)

### 4.4 Reproducible Research

The concept of Reproducible Research was born to deal with the problem that is reproducing results of a paper. Usually, when the work presented in a written form is read, it is very hard for the reader to replicate the results himself. The parameterization, the input files, the code, the machines used, are all variables that can change the output.

With this in mind, Patrick Vandewalle [Ven10] introduced this idea in his website. According to the author, this methodology has the following advantages:

1. It helps the author the work to reproduce early results in latter stages of the research.
2. Other people that want to carry on with the research can start with the current state of the art, instead of spending months trying to achieve the same results as the paper.
3. It simplifies the comparison between methods, assuring the right implementation of the methods presented.

In order to make a paper reproducible, there are some recommendations that should be followed. Starting with a good description of the algorithm and experiments on the paper, a website should be created containing the following:

1. Title of the work
2. Authors (and respective websites)
3. Abstract
4. Full reference of the paper with publication status and PDF of the complete text.
5. All the data used to produce the results with a readme file to explain what the data represents

6. A list of configurations on which the code was tested (software, hardware)

7. An email address for comments and bug reporting

8. References (with abstracts, if possible)

This work, being a study on several binarization methods, will provide the code for all the methods implemented with comments, all the images of the dataset, the ground truth binarizations used for the ME, and a script that calls all the methods in sequence, to produce all the thresholded images and results.

The standardization of the implementations mentioned before in this chapter plays an important part, assuring similar formats in inputs and outputs of different algorithms thus making it easier to reproduce the results.
Methodology, Evaluation Metrics and Results
Chapter 5

Conclusion and Future Work

This chapter presents the conclusion of the work, describing the completed objectives and future work that can be done in the area.

5.1 Objectives Completed

This project presented an evaluation and analysis of different binarization methods, applied to music score binarization, which has been an overlooked aspect in OMR research.

A new method (BLIST) for image binarization that makes use of content that is common to music scores. As a global method, BLIST has better results for music scores that any of the state of the art methods tested, failing only in the cases where it is impossible for global methods to perform perfectly (in some scores, the method either assures the presence of the staff lines or removes all the occlusive noise).

An adaptive version of this method was also implemented, by applying it independently to vertical sections of the image and interpolating the results, producing a threshold value for each vertical line of pixels. This Adaptive BLIST, although producing less noise and almost no occlusion of information, also damages the staff finding process slightly and therefore may not justify the increase in computational cost that is common for adaptive methods.

5.2 Future Work

This project opens some future work perspectives for further research.

The adaptive version of the BLIST method can be improved to produce results that are near the best possible, providing a complete removal of noise while keeping the good
Conclusion and Future Work

results in the staff finding process. One possible way that can guide this work is to some-
how treat the left and right margins of the scores differently, as most of the times they do not carry information but still affect the end result.

Another way to treat this problem is to use global BLIST as a binarization method for the staff finding process, while defining a new method for the symbol recognition steps that come next. Although it is not common for OMR systems to used more than one binarization process, that approach is worth being considered.

Music scores are not the only type of document that can benefit from content aware binarization. A good example of a way to apply the ideas presented in this report would be to find the character stroke thickness in documents that are mostly text, similarly to what BLIST does for the music staves. The concept of content aware binarization, however, is not exclusive to methods that find thickness of lines in documents, and can use any information that is characteristic to a type of image.
References

[Abu89] A. S. Abutaleb. Automatic thresholding of gray-level pictures using two-


REFERENCES


REFERENCES


REFERENCES


