ATHENA: Automatic Text Height ExtractioN for the Analysis of text lines in old handwritten manuscripts

Ruggero Pintus, Yale University, USA / CRS4, Italy
Ying Yang, Yale University, USA
Holly Rushmeier, Yale University, USA

Massive digital acquisition and preservation of deteriorating historical and artistic documents is of particular importance due to their value and fragile condition. The study and browsing of such digital libraries is invaluable for scholars in the Cultural Heritage field, but requires automatic tools for analyzing and indexing these datasets. We present two completely automatic methods requiring no human intervention: *text height estimation* and *text line extraction*. Our proposed methods have been evaluated on a huge heterogeneous corpus of illuminated medieval manuscripts of different writing styles and with various problematic attributes, such as holes, spots, ink bleed-through, ornamentation, background noise, and overlapping text lines. Our experimental results demonstrate that these two new methods are efficient and reliable, even when applied to very noisy and damaged old handwritten manuscripts.

Categories and Subject Descriptors: I.7.5 [**Document and Text Processing**] Document analysis; I.7.5 [**Document and Text Processing**] Graphics recognition and interpretation

General Terms: Document Layout Analysis, Cultural Heritage

Additional Key Words and Phrases: Handwritten manuscripts, Layout analysis, Text height estimation, Text line segmentation, Frequency based analysis

ACM Reference Format:

Ruggero Pintus, Ying Yang, Holly Rushmeier. 2014. ATHENA: Automatic Text Height ExtractioN for the Analysis of text lines in old handwritten manuscripts. *ACM J. Comput. Cult. Herit.* 0, 0, Article 0 (January 2013), 23 pages.
DOI:http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Books and manuscripts are being digitized at an increasing rate. In Cultural Heritage this digitization activity becomes particularly important. A huge number of historical and artistic documents are deteriorating day by day, and their digital preservation is required due to their value and fragile condition. Moreover, a digital collection of such documents represents an invaluable database that would not otherwise be available to the public, whether they are experts, tourists or people keen on art. The amount and importance of the information contained in this variety of different language manuscripts motivates the development of tools to explore, read, and enjoy them in a more comprehensive manner.

Address: Yale Graphics Lab, Department of Computer Science, 51 Prospect Street, New Haven, CT 06520, US; email: ruggero.pintus@yale.edu, ying.yang.yy368@yale.edu, holly.rushmeier@yale.edu; web: http://graphics.cs.yale.edu

DOI:http://dx.doi.org/10.1145/0000000.0000000

Digital libraries from cultural institutions all over the world have yet to be fully exploited for consulting, exchange, remote access and textual search. Dealing with massive databases of thousands of pages requires automatic techniques to perform per-page analysis and classification. Thus, document layout analysis plays a significant role, being a fundamental step of any document image understanding system. Although some efficient algorithms have been proposed to cope with modern machine-printed documents or historical documents from the hand-press period, analyzing old handwritten manuscripts poses additional challenges. These documents (e.g., illuminated medieval manuscripts) are seriously degraded due to ageing, and greatly differ from the modern machine-printed documents in layout and formatting. Further, their physical structure, containing text, capital letters, portraits, ornamental bands and graphical contents, is even harder to extract due to numerous uncontrollable factors, such as holes, spots, writing from the verso appearing on the recto (ink bleed-through), ornamentations, background noise, touching text lines, different writing styles, and so forth. The segmentation of historical handwritten documents is still an open research problem, and, to the best of our knowledge, a completely automatic solution to it has not been reported.

A common problem in document structure analysis is the initial estimation of the text height. While there are some good automatic techniques to find this value for printed documents with clear and substantial inter-line spacing, the estimation of text height becomes particularly challenging as the inter-lines become narrower, and when descenders and ascenders start to touch each other and to fill the blank space between adjacent text lines (e.g., in medieval manuscripts). In such difficult cases, due to the ease of manually performing this task, some state-of-the-art techniques [Mehri et al. 2013; Garz et al. 2012; Journet et al. 2008] require the user to input a rough estimation of the text height, which is then used to define some algorithm parameters. However, user intervention becomes infeasible when dealing with massive datasets that contain a high number of different documents and high variability of text size from one manuscript to another, within the same manuscript, or, even worse, across a single page or text line. Moreover, applied as a preprocessing step, an automatic estimation of the text height would make some state-of-the-art existing algorithms completely automatic. For instance, the initial computation of the text height value is crucial in text line extraction, which is another important topic in document understanding. Although it has been seen generally as a preprocessing step for layout extraction, character or word spotting, or handwriting recognition, text line identification is broadly used as a standalone task, and as a fundamental tool to assist scholars in manuscript transcriptions [T-pen 2013].

This paper is a significant extension of our Digital Heritage 2013 contribution [Pintus et al. 2013], which presented a reliable and fast approach to perform an automatic text height estimation. Throughout this paper we consider the text height as a synonym of text leading. In that paper, given the image of a manuscript page, a multi-scale representation is first produced. Then, for each sub-image at each level, a new, robust descriptor is computed; this is based on a frequency analysis of the y-axis projected profile of the normalized image autocorrelation function. Finally, by exploiting spatial consistency between the proposed image descriptors at different scale levels, a voting procedure finds the predominant spatial frequency in the document page, whose period dimension is the value of the text height. We showed that this new method is efficient and reliable, even for very noisy and damaged old handwritten manuscripts; we demonstrated its efficiency on a huge heterogeneous corpus content with different writing styles, text sizes, image resolutions, and levels of conservation.

Besides supplying a more thorough exposition and evaluation of the text height extraction algorithm, we provide here significant new material. The main motivation for this extended version is to show how our previous contribution works with a broader kind of input data (e.g., different languages and writing styles), and how it could be exploited by integrating it in more general, automatic document layout analysis frameworks. We present here a robust, parameter-free, automatic pipeline for a per-page basis text line segmentation. Given the image of a manuscript page, we first exploit the frequency-based descriptor introduced by Pintus et al. [2013] to obtain an automatic coarse segmentation mask of text regions. We refine this segmentation by combining the computed text height value, robust image features (e.g., SIFT [Lowe 2004]), and machine learning approaches based on Support Vector Machine (SVM) models. A reliable estimation of pixels belonging to text areas is integrated into a commonly used framework for extracting lines of text, in order to decrease the chance of false positives or negatives. Our main contributions are the following:

Automatic coarse text segmentation. We introduce a new operator to perform a coarse segmentation of areas containing text (see Section 6.1). It relies on the same frequency-based descriptor presented by Pintus et al. [2013] applied here to each pixel at different window sizes; these sizes are automatically defined by the estimated text height value.

Automatic per-page basis text line extraction. We present a modification of the commonly used Projection Profile framework, to obtain an automatic extraction of text lines on a per-page basis (see Section 6.2). We exploit the automatically computed coarse text segmentation, the text height value, image descriptors, and an SVM classifier with Gaussian radial basis function, to select representative text pixels only, and use them to perform an adaptive integration along image rows.

Evaluation. We extend the text height extraction evaluation by presenting new results obtained for different languages (e.g., non-Latin documents) and writing styles. To assess our new contribution we present a detailed and extensive evaluation of the proposed algorithm for text line extraction, applied to a large and heterogeneous corpus content with different inter-line distances, noise, ornamentation and illumination, image resolutions, and levels of conservation.

2. RELATED WORK

Document analysis is one of the most well studied fields in image processing. A huge amount of work has been presented to deal with different aspects ranging from segmentation [Grana et al. 2009], line extraction [Jindal and Lehal 2012], char and word spotting [Diem and Sablatnig 2010; Yalniz and Manmatha 2012], and classification of handwritten documents and medieval manuscripts [Louloudis et al. 2008; Leydier et al. 2007]. An exhaustive review of the literature is far outside the scope of the paper, and the reader is referred to the seminal work of Nagy [Nagy 2000], which gives an overview of early techniques proposed for text segmentation, OCR and background removal, and various recent surveys [Sharma et al. 2012; Likforman-Sulem et al. 2007]. Here we discuss only the state-of-the-art techniques closely related to ours.

Integration profiles. Commonly used approaches to determine text height estimation or text line segmentation are based on Projection Profiles [Bulacu et al. 2007; Shapiro et al. 1993; Antonacopoulos and Karatzas 2004], the XY-CUT algorithm [Khedekar et al. 2003], and the Run Length Smearing Algorithm (RLSA) [Wang et al. 2002]. They are all based on different ways to directly integrate the original image along rows, columns or, rarely, diagonal directions. Based on strong a priori assumptions, they have two common main drawbacks: short lines will produce weak signal, and very narrow lines with overlapping descenders and ascenders will not provide significant signal at all. While these approaches are mainly used for printed documents, some papers adapted them to handwritten ones with little overlap between lines and moderately skewed texts [Bar-Yosef et al. 2009; Zahour et al. 2001]. It follows that, although these solutions are typically faster, they are very sensitive to noise, and not robust enough to be directly applicable to a generic handwritten, possibly damaged medieval manuscript, with generic layout rules, irregularities in script and writing style, skew, overlapping and fluctuating text lines. Further, they are not completely automatic approaches, because they require the user to input some manually defined or text style dependent parameters to avoid local minima in the analysis of projection profiles [Jain and Namboodiri 2003; Ratzlaff 2000].

Local descriptors. Recent works perform handwritten text and pattern characterization by extracting more robust orientation-based features, such as histograms of oriented gradients [Minetto et al. 2012], Gabor descriptors [Eglin et al. 2007], scale invariant features (e.g., SIFT) [Garz et al. 2010], and an autocorrelation function [Mehri et al. 2013; Journet et al. 2008], which allow them to analyze the document layout, and estimate similarities and differencies between its regions without any hypothesis about its physical and logical structure. Moreover, from the autocorrelation function signal one could obtain the so called Rose of Direction (RoD) [Journet et al. 2005], which makes it possible to apply a well known local or global skew correction, as an optional additional preprocessing step. The main issue is that all mentioned techniques require user intervention, either to train some classifiers [Garz et al. 2010], or to manually set some parameters that are strictly dependent on the document text height, such as in the case of the neighborhood radius in Garz et al. [Garz et al. 2012], or the size of kernel windows in Mehri et al. [Mehri et al. 2013]. Although these solutions are very robust to noise and can cope with non-idealities in the input data, manually adjusted parameters limit the range of their applicability, and make them unsuitable for a massive, non-homogeneous corpus with different acquisition resolutions, writing styles and text heights.

Multi-scale representations. Exploiting a multi-resolution representation and a frequency-based framework is an old and well-known approach in image analysis and segmentation (e.g., [Sabharwal and Subramanya 2001]), which has been applied to a plethora of applications; generally speaking, it is used to treat in an adaptive way different kinds of input data, e.g., in terms of resolution and size of image details. In the specific field of document layout analysis these methods are typically used to segment document images scanned from newspapers and journals [Qiao et al. 2006; Lemaitre et al. 2008]. Recently, Almeida et al. [Almeida and Almeida 2012] used wavelets to reduce ink show-through noise in scanned letters or images. Joutel et al. [Joutel et al. 2008] presents a multi-level curvelets decomposition of ancient document images for indexing linear singularities of handwritten shapes; it allows for applications such as manuscripts dating, expertise and authentication of its author, style and period.

Our contribution. The two techniques presented here were inspired by the aforementioned works, aiming at attaining a completely automatic pipeline for text height estimation [Pintus et al. 2013] and text line segmentation (novel contribution).

In the text height extraction algorithm, instead of relying on projection profiles directly obtained from the original image, we compute the y-axis profile of the more robust normalized autocorrelation function, which proves to be reliable in the presence of noise and other factors such as ink bleed-through, ageing and damages in old manuscripts. Our method is independent of document brightness and contrast, and skewed text. Instead of defining some parameter values to deal with local maxima and minima in the profile, we analyze it by extracting its discrete Fourier coefficients, and by estimating the most predominant spatial frequency in a parameter-free manner; this results in a text height estimation that is robust to noise in the projection profile as well. A complete and reliable automatic solution is achieved by integrating this local image representation into a multi-scale framework, where a descriptor is computed at different scale levels.

Additionally, we introduce a new technique for automatic text line extraction. Although advanced methods for text line extraction exist for extreme cases, such as skewed and non-rigid deformed text [Koo and Cho 2010], handwritten Arabic lines [Shi et al. 2009], and chinese characters [Koo and Cho 2012], all of them deal with datasets with high contrast between background and foreground, so that noise is not a problem. As an extended version of our previous paper [Pintus et al. 2013], our purpose here is to show how our frequency-based descriptor and the evaluation of the text height can be easily employed within standard document layout analysis approaches. For this reason, a deep and extensive comparison between this text line segmentation algorithm and the state-of-the-art literature is far out of the scope of this paper, and will be addressed in future work. Particularly, we propose here a modification of the commonly used Projection Profile framework, where we integrate these tools to provide an automatic, parameter-free extraction of text lines on a per-page basis.

Finally, an extended and extensive evaluation is performed, proving the robustness and reliability of both the presented methods, and showing that they are well-suited for huge digital libraries with a high variability of layouts, syles, languages, and levels of conservation.

3. TECHNIQUE OVERVIEW

Fig. 1 and Fig. 2 show the pipelines of the two proposed techniques.

Text height extraction (Fig. 1). The algorithm is given as input an image of a manuscript page that, without loss of generality, contains a quasi-horizontal text; otherwise, we could apply, as an additional pre-processing step, the well-known automatic page alignment correction based on the Rose of Directions approach [Journet et al. 2005]. First we produce an N-level multi-scale representation; at level n, we split each original image in 2^{2n} small sub-images. Then, we analyze these levels separetely. For each of their sub-images we compute the normalized autocorrelation function (NACF), and we integrate this signal to obtain its y-axis projection profile (y_{pp}). We find the main periodicity of the y_{pp}P by applying the Discrete Fourier Transform (DFT). We use the information corresponding to the highest DFT coefficient from all sub-images to compute, for that particular level n, an estimation of the text height in terms of probability mass function (PMF). Finally, we exploit the coherence between levels to find the final estimation of the page text height, by accumulating all the PMFs from all levels.

Text line segmentation (Fig. 2). The input data is an image of a manuscript page containing text, and the value of the estimated average text height (Fig. 1). First of all, we extract image keypoints, and the corresponding descriptors from the given image. In parallel, by exploiting the information of the text height value and by using a frequency-based operator, we perform a coarse text region segmentation. For each keypoint we try to estimate its compatibility with a text feature; we do this by comparing its radius (i.e., diameter of its meaningful neighborhood) with the text height value, and by checking that its position is inside the coarsely segmented region. Accordingly, we assign a label to each of them (i.e., *text* or *non-text*), in order to define whether or not they belong to a text region and are highly representative of a text character shape. This rough labelling is employed to train a robust SVM classifier with Gaussian radial basis function kernel. Launching the resulting prediction on all the original image descriptors we will obtain a refined text region segmentation, sampled at the keypoints location. Finally we compute the projection profile by integrating only the contribution of the text keypoints. The size of profile bins, the algorithm to extract the maxima of the profile, and the final segmentation of each single line, all are strongly dependent of the previously computed text height value.

4. INPUT DATA

In general, the nature of the input data in the presented algorithms is unconstrained; the only requirement is that it is an image of a manuscript page containing text. It can have figures, ornamentation, capital letters, portraits, touching and overlapping texts, and can be degraded by background noise, ink bleed-through and other kinds of damage due to ageing. The only mild assumption is that either the acquisition setup is such that the text is quasi-horizontal, or that a pre-processing step is applied in order to correct the overall page orientation. This could be easily done by employing the well-known Rose of Directions method [Journet et al. 2005] or more recent, powerful techniques [Papandreou et al. 2013]; here, in section 7, we also present an additional alternative solution to correct orientation. In our case, however, since we use operators that are very robust to skewed texts, we will see how this is a very relaxed constraint, and how typical acquisition setups do not require any alignment correction at

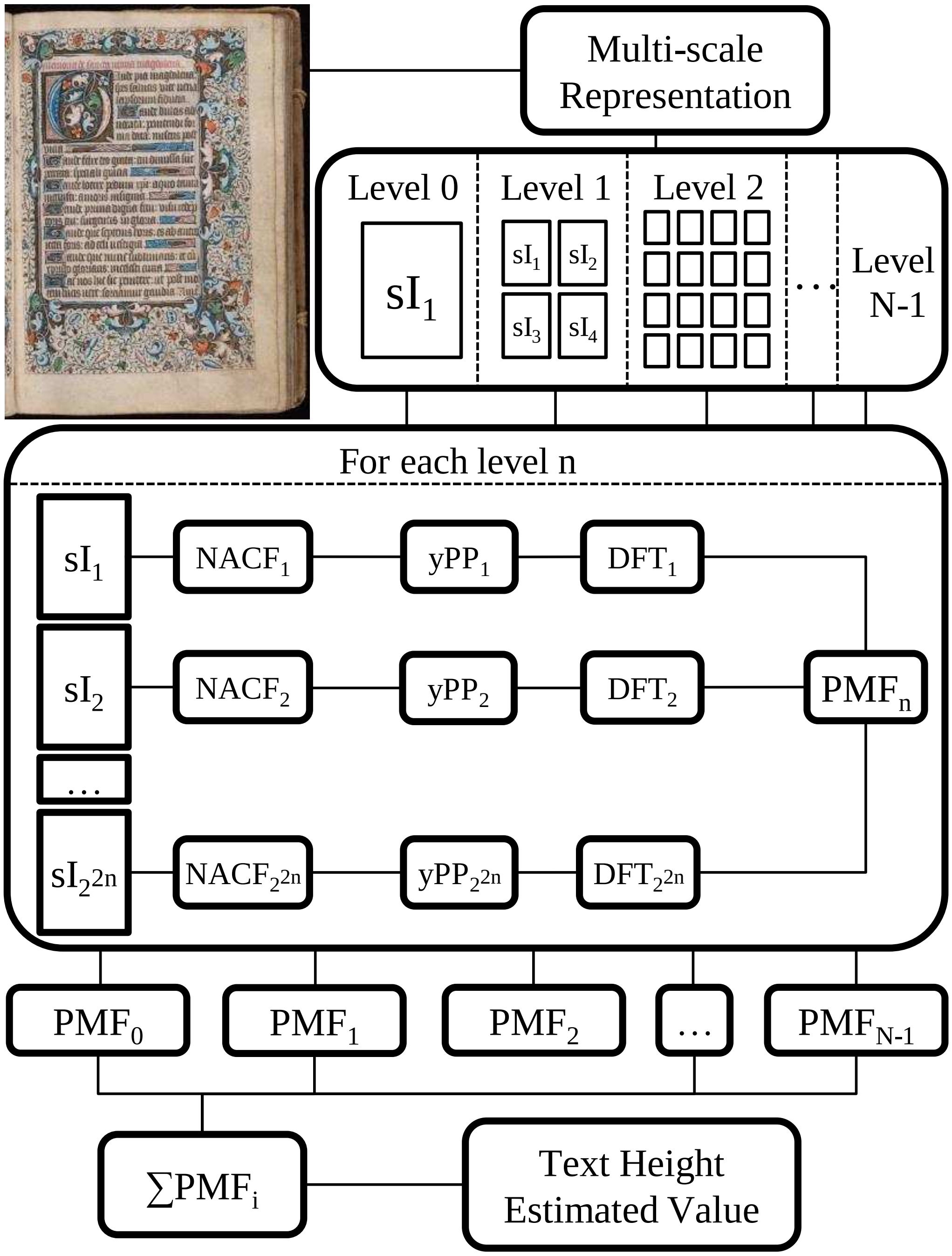


Fig. 1. **Text height extraction - Algorithm pipeline.** Given an input image we compute its multi-level representation. After estimating the text height Probability Mass Function (PMF) for each level, we obtained the final estimation by a voting framework across all levels. Manuscript image courtesy of the Yale University[BeineckeMS310].

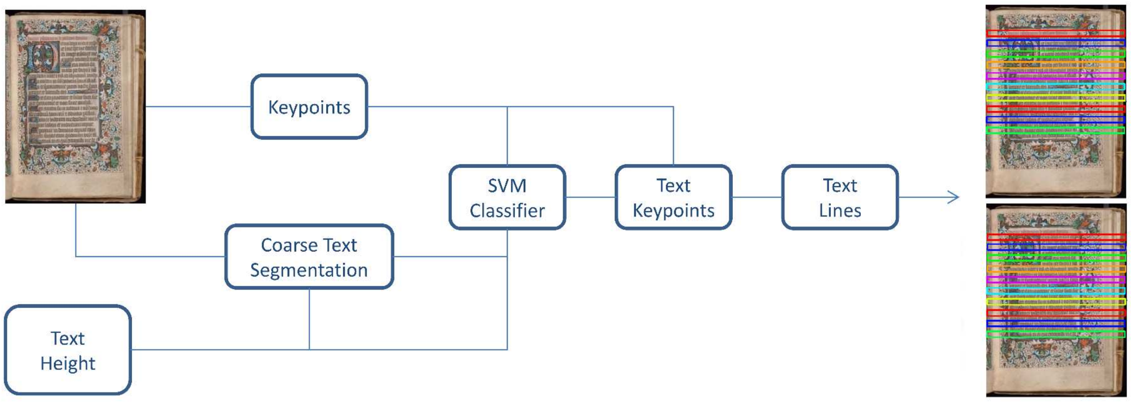


Fig. 2. **Text line segmentation - Algorithm pipeline.** Given an input image, we extract image keypoints and perform a coarse text region segmentation. We use the text height value, the radii of keypoints and the coarse segmentation to produce a rough feature labelling (i.e., *text* or *non-text*) and to train a robust SVM classifier. The fine segmentation of keypoints from the SVM prediction is employed in a projection profile framework to obtain the final text line segmentation. We show two images with the retrieved odd and even lines of text. Manuscript images courtesy of the Yale University[BeineckeMS310]

all. Hence, to produce the results in section 8, we do not use any orientation correction. We don't need any a priori information about the language. However, we need to know the main orientation of the text in the book, whether it is horizontal (e.g., Latin) or vertical (e.g., traditional Chinese). In the case of vertical orientation, we adapt the algorithm by rotating the image 90 degrees before launching the process. This information might be a metadata, which can be easily provided with each manuscript. Another assumption for the text line segmentation is that the page contains only one text block.

5. TEXT HEIGHT ESTIMATION

In this section we explain in detail the text height extraction technique; for illustration purpose only, we use the sample image in Fig. 3(a) to show all the steps of our approach.

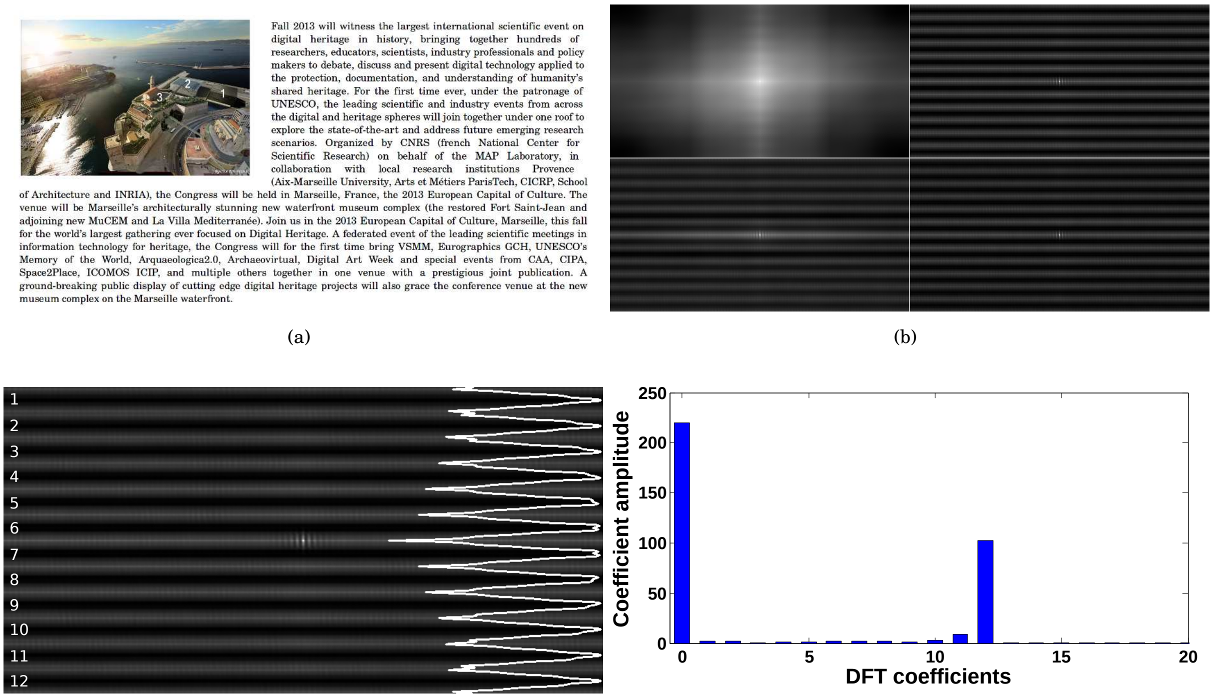
5.1 Multi-scale representation

First, we compute a multi-scale representation of the input image. Considering a particular level n we split the original image in 2^{2n} small sub-images. The number of levels must be fixed; it must be uncorrelated with the acquisition resolution, and independent of the text height, the layout and the structure of the manuscript page. We choose this value by relying on the following considerations.

On one hand, since multi-scale analysis is based on consistencies across different levels, it would seem obvious that the more levels, the more robust the algorithm. On the other hand, given an arbitrary high level value, the probability that a sub-image contains one or more text lines tends exponentially to zero. The sub-images that do not contain any spatial periodicity are discarded (as we will see in detail in the section 5.4) for computational efficiency. We need to choose the number of levels both as reasonable, general and conservative. With these considerations, and the fact that we are dealing with handwritten texts, we empirically found that a 5-level multi-scale representation is a reliable parameter value in all our extensive tests.

5.2 Single level analysis

After building the multi-scale representation, we perform a separate analysis of each level. We start by computing the normalized autocorrelation function of each sub-image at that level. The autocorrelation



Fall 2013 will witness the largest international scientific event on digital heritage in history, bringing together hundreds of researchers, educators, scientists, industry professionals and policy makers to debate, discuss and present digital technology applied to the protection, documentation, and understanding of humanity's shared heritage. For the first time ever, under the patronage of UNESCO, the leading scientific and industry events from across the digital and heritage spheres will join together under one roof to explore the state-of-the-art and address future emerging research scenarios. Organized by CNRS (french National Center for Scientific Research) on behalf of the MAP Laboratory, in collaboration with local research institutions Provence (Aix-Marseille University, Arts et Métiers ParisTech, CICRP, School of Architecture and INRIA), the Congress will be held in Marseille, France, the 2013 European Capital of Culture. The venue will be Marseille's architecturally stunning new waterfront museum complex (the restored Fort Saint-Jean and adjoining new MuCEM and La Villa Mediterranée). Join us in the 2013 European Capital of Culture, Marseille, this fall for the world's largest gathering ever focused on Digital Heritage. A federated event of the leading scientific meetings in information technology for heritage, the Congress will for the first time bring VSMM, Eurographics GCH, UNESCO's Memory of the World, Arquaeologica2.0, Archaeovirtual, Digital Art Week and special events from CAA, CIPA, Space2Place, ICOMOS ICIP, and multiple others together in one venue with a prestigious joint publication. A ground-breaking public display of cutting edge digital heritage projects will also grace the conference venue at the new museum complex on the Marseille waterfront.

(a)

(b)

(c)

(d)

Fig. 3. **Frequency-based descriptor.** Given a sample image (a), for each sub-image at each sub level, we compute the normalized autocorrelation function (NACF)(b). We show the NACF integration along x axis to obtain the y-axis projection profile signal y_{pp} of the top-right sub-image at level 1 (c), and its discrete fourier coefficients (d). Sample image (a) was taken from the Digital Heritage 2013 conference website[DigitalHeritage]

function for a two dimensional signal is defined by:

$$ACF\left(x,y\right) = \sum_{\alpha\in\Omega}\sum_{\beta\in\Omega} I\left(\alpha,\beta\right) I\left(\alpha+x,\beta+y\right) \tag{1}$$

The autocorrelation value at position (x,y) is the sum of the products of the grayscale image values $I(\alpha,\beta)$ and the pixel values after a translation of (x,y). These translations are at the basis of the inspection of the input sub-image according to its different directions. The normalized autocorrelation function $(NACF)$ is:

$$NACF\left(x,y\right) = \frac{ACF\left(x,y\right) - min_{ACF(x,y)}}{max_{ACF(x,y)} - min_{ACF(x,y)}} \tag{2}$$

where min and max are the minimum and maximum values of the autocorrelation function. Fig. 3(b) shows the normalized autocorrelation functions of sub-images at level 1. We can clearly see the difference between sub-images that contain text lines or figures.

To extract the spatial periodicity of the patterns that correspond to text regions, we compute the y-axis projection profile $y_{pp} = YPP\left(NACF\right)$ of the $NACF$. In Fig. 3(c) we superimpose the $NACF$ and the profile (white curve) of the top-right sub-image at level 1. We analyze its frequency footprint by computing the DFT coefficients. After discarding the constant component (i.e., 0-index coefficient), the coefficient with the highest amplitude corresponds to the predominant spatial frequency. In other words, if the coefficient with the highest amplitude has index i_{max}, it means that the signal has i_{max} periods inside the studied domain. At each level n, we define A_n^{χ} the maximum coefficient amplitude for the sub-image χ. After computing the DFT of the profile in Fig. 3(c), in Fig. 3(d) we plot the amplitude of the first 100 coefficients. The $12th$ coefficient has the highest amplitude, i.e., the profile in Fig. 3(c) has 12 periods. For each sub-image, the size in pixels of that period, obtained by dividing the sub-image height by the number of periods, is a possible candidate value for the text height estimation at that particular level.

Now we have to merge the information of all the 2^{2n} sub-images. In a histogram we accumulate the amplitude of the 2^{2n} most relevant coefficient; each of them is the coefficient with the maximum amplitude from the corresponding sub-image. Then, in this histogram, the index with the highest amplitude integral is the winner for the current level. For instance, in Fig. 3(b), those amplitudes fall into histogram bin 12, while there is only a single amplitude in bin 1. However, due to the discrete nature of the performed analysis, we do not want to produce a single level value for the text height estimation. On the other hand, for each level n, we prefer to build the following Gaussian probability mass function (PMF) of the text height random variable t:

$$PMF_n\left(t\right) = w_n \cdot e^{-\frac{1}{2}\frac{(t-\mu_n)^2}{\sigma_n^2}} \tag{3}$$

$$\mu_n = \frac{th_n^{min} + th_n^{max}}{2}, \sigma_n^2 = \left|th_n^{max} - \mu_n\right|^2 \tag{4}$$

where $th_n^{min} = height_n / \left(i_n + 0.5\right)$, $th_n^{max} = height_n / \left(i_n - 0.5\right)$, i_n is the winner coefficient index, and $height_n$ is the height of the level sub-image. The level-based normalized weight w_n:

$$w_n = \frac{1}{C_n \cdot width_n} \sum_{\chi=1}^{C_n} A_n^{\chi} \tag{5}$$

serves to make the PMFs from different levels comparable. C_n is the number of sub-images at level n whose coefficients are equal to the winner index i_n, and $width_n$ is the width of sub-images at level n.

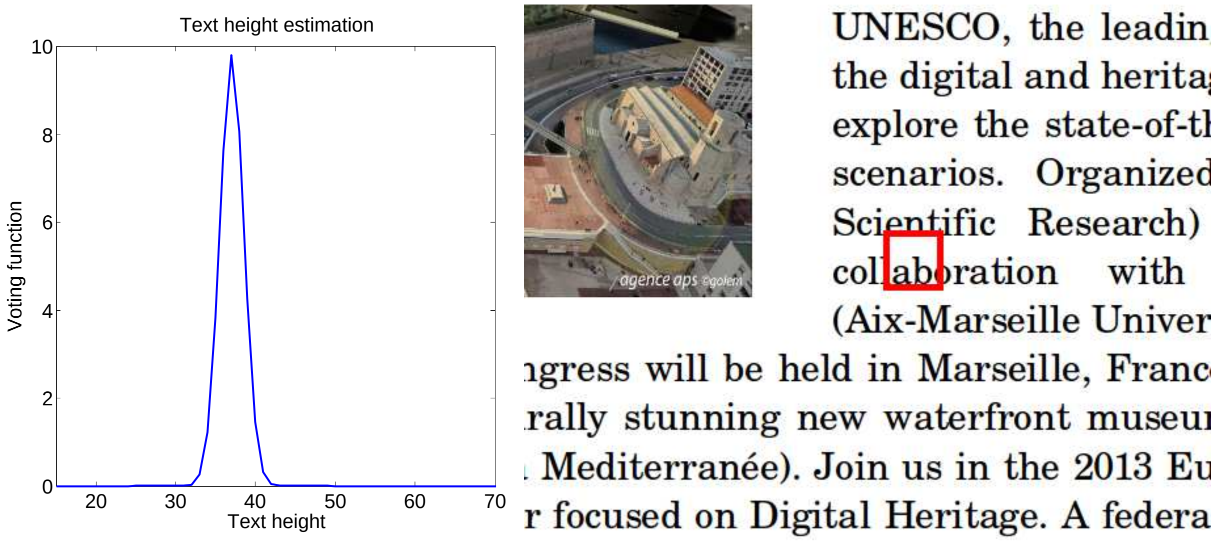
5.3 Multi-level analysis

The result of the previous step is a set of N Gaussian probability mass functions, one for each level. Each one, with its mean value and variance, gives a per-level estimation of the possible text height value for the analyzed page. We want to combine all these PMFs, considering the property that sub-images containing text at different levels produce similar expected values, even if the number of periods or the corresponding amplitudes are different. In order to exploit this consistency between levels, we employ a voting framework in which we compute a voting function by accumulating all the PMFs. The value t_E corresponding to the maximum of this function is the final text height estimation for the manuscript page. Fig. 4(a) shows the multi-level voting function obtained by accumulating PMFs from the sample image in Fig. 3(a). To validate this estimation, in Fig. 4(b) we show a zoomed region of the sample image, in which we draw a square with the edge size equal to the corresponding estimated t_E.

5.4 Implementation

In this section we describe some implementation details that make the proposed method more robust and efficient.

First of all, the direct computation of the autocorrelation function as expressed in equation 1 is computationally inefficient. However, we can use the Plancherel theorem, which allows us to more



UNESCO, the leadin
the digital and herita
explore the state-of-t
scenarios. Organized
Scientific Research)
collaboration with
(Aix-Marseille Univer
ngress will be held in Marseille, Franc
rally stunning new waterfront museu
Mediterranée). Join us in the 2013 Eu
r focused on Digital Heritage. A federa

(a) Voting function

(b) Estimated text height

Fig. 4. **Multi-level analysis.** A voting function is obtained accumulating all the N Gaussian probability mass functions from all levels (a). The height corresponding to the maximum of this function is the estimated text height (t_E). To check the quality of the algorithm outcome, a square with edge size equal to t_E is drawn over the original text (b). Sample image (b) was taken from the Digital Heritage 2013 conference website[DigitalHeritage]

efficiently express the equation in terms of the image Fourier transform [Mehri et al. 2013]:

$$ACF\left(x,y\right)=FFT^{-1}\left[FFT\left[I\left(x,y\right)\right]FFT^{*}\left[I\left(x,y\right)\right]\right] \tag{6}$$

where FFT is the Fast Fourier transform operator, FFT^{-1} is its inverse and FFT^{*} is its complex conjugate.

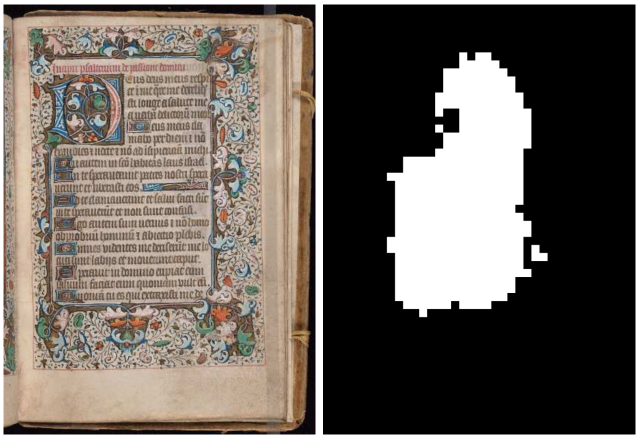
We have found that the contribution of the level 0 to the computation of the final text height is generally very poor; it would be useful with huge text heights (e.g., text height bigger than half image height size), but is a very rare scenario we never encountered in our database. Since its analysis is the most computationally expensive, by discarding that level we obtain a significant speed up without changing the output result.

Based on the properties of the normalized autocorrelation function and its y-axis profile, we can apply an outlier pruning strategy in the single level PMF computation step. Image parts that contain figures do not have a main direction, so their NACF is typically a homogeneous signal with a high value at the center pixel; its profile is a curve with one high central peak, and a decreasing behaviour as a function of $\left|\frac{1}{x}\right|$. In these cases, the index of the most relevant DFT coeffient is 1 (index 0 is the constant coefficient), i.e., one period in the studied domain. Since we are looking for spatial periodicities, we avoid accumulating all these coefficients with index $i_n \leq 1$.

We have found that these implementation choices result in a big improvement both in text height estimation reliability and in the computational efficiency.

6. TEXT LINE SEGMENTATION

The input data of the text line segmentation pipeline consists in an image of a manuscript page (see Section 4 for further details), and the value in pixels of the estimated average text height for that page. As preprocessing steps, we first extract image keypoints and their corresponding descriptors,



(a) (b)

Fig. 5. **Automatic coarse text segmentation.** a) the original image of a manuscript page; b) the coarsely segmented text obtained with the proposed new operator, which relies on the frequency-based descriptor presented by Pintus et al. [2013]. Manuscript image courtesy of the Yale University[BeineckeMS310].

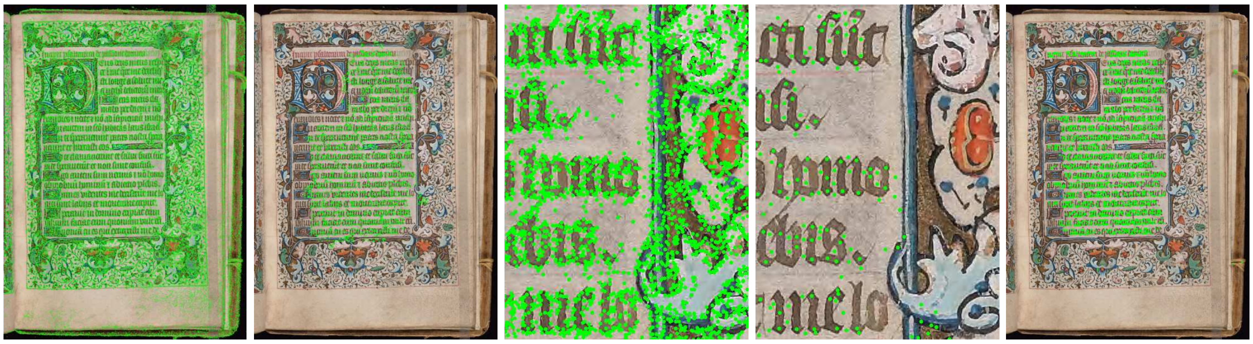
and then we perform an initial coarse segmentation of text regions in the page. In this work we use SIFT features (Scale-Invariant Feature Transform) [Lowe 2004].

6.1 Automatic coarse text segmentation

Both the knowledge of the average text height value t_E and the frequency-based analysis introduced in section 5 allow us to build an operator to coarsely estimate the likelihood of a pixel to belong to a text part of a manuscript page. The main insight behind text segmentation is the following.Given a pixel, we consider a window of size nt_E centered at that pixel, and we compute the y-axis projection profile of its normalized autocorrelation function (i.e., y_{pp} as in section 5). After it performs a frequency analysis of that profile, if the pixel happens to lie in a text region, the maximum coefficient index i of the Discrete Fourier Transform (DFT) will likely be equal to n. Using only one window, the segmentation is not robust enough, so we do the same check with a broader range of integers n in the interval $[2, N]$. With the aim at training a classifier, we need a very conservative initial identification of text pixels. For this reason, each pixel that meets the constraint $i = n$ for *all* the window sizes will be marked as *text* pixel, otherwise we marked it as *non-text*. We heuristically found $N = 5$ a reasonable choice in terms of reliability and computational time, and we used it for all the presented results. In Fig. 5 we show the original image of the page, and the mask after the coarse segmentation; white pixels are labelled as *text*, while black as *non-text*.

6.2 Automatic extraction of text lines

Our algorithm to extract the lines of text in the manuscript page relies on a fine, text-based segmentation of the image features computed in the pre-processing step. We aim here at finding the most representative keypoints that belong to text characters. The algorithm is given all the SIFT keypoints for a particular image, and we perform a labelling of those by taking into account the information we have so far, i.e., the text height value and the coarse mask. For each keypoint we consider its position in the image and its scale. We mark it as a *non-text* keypoint both if it lies outside the *text* region of the mask, and if its scale is not compatible with the size of the text height. Otherwise, we mark it as a *text* keypoint. More precisely, since a feature generally describes only a part of a character, we found that the most salient keypoints typically have a size between the 5% and the 25% of the text height; we choose these fixed constraints in all of our experiments. In fig. 6(a) we show in green the positions of all the extracted, original image features, while in fig. 6(b) are displayed only those after the pruning; in a more detailed view (fig. 6(c) and fig. 6(d)) it is possible to see how the algorithm mainly keeps the most relevant keypoints related to text characters, such as the edges of each single vertical stroke, vertical background space between strokes, or other peculiar text shapes, while it discards the majority of noisy features from the background (noise in the parchment), and keypoints from pictures, ornaments or other non-textual parts.



(a) Original SIFT keypoints

(b) Pruned SIFT keypoints

(c) Original SIFT keypoints - A detail

(d) Pruned SIFT keypoints - A detail

(e) Text Keypoints - Fine segmentation

Fig. 6. **Text features.** The green dots represent the position of keypoints in the image: a) original keypoints; b) pruned keypoints that lie in the coarsely segmented region, and have a scale compatible with the size of the text height; c) and d) zoomed regions showing the pruning operation. Finally we show the refined text feature classification in e). Manuscript images courtesy of the Yale University[BeineckeMS310].

However, the resulting keypoint labelling is a rough classification that could contain a certain number of outliers (both false positives and/or false negatives); this is due to the heuristic and conservative nature of the allowed feature scale range, and to the previously computed coarse segmentation of text regions. In our case, we found that the presence of outliers makes the use of linear kernels not suitable for our problem. Conversely, a Support Vector Machine model (SVM) with Gaussian radial basis function as kernel proved to be able to better separate the two classes (i.e., *text* and *non-text*) in the SIFT feature vector space. We use a *two class C-Support Vector Classification*, with a γ value of the gaussian kernel equal to $\frac{1}{2\sigma^2}$, where σ^2 is the variance of the training dataset. Once the SVM is trained, we produce a refined feature classification by simply launching the prediction over all the original SIFT descriptors. Then we keep only the keypoints marked as *text* by the classifier. We show the fine segmented features in Fig. 6(e). The improvement of the refinement step is clearly visible in some parts; for instance it recovers text keypoints in the first line (the reddish text), and gets rid of those in non-text regions, such as the side of the big illuminated *D* champ initial, and the horizontal blue and red bar on the right of the 11th line.

At this point, in order to extract the text lines, we modify the original projection profile algorithm [Likforman-Sulem et al. 2007] in two ways: we integrate only the contribution of *text* keypoints; rather than considering a one-pixel resolution, we set the bin size of the profile proportional to the known text height. In fact, in the case of the integration of a sparsely sampled field (the set of keypoint positions), too small a bin size (e.g., one pixel) will result in a potentially noisy signal, while too big a value will have a poor resolution, and the information about text lines will be lost. We set this size as a quarter of the text height, and use this value for all our tests. Finally, we find all the maxima in the profile.

Some state-of-the-art methods solve this by providing a user defined tolerance to avoid local maxima. Instead, in a parameter-free context, we assign the text height value to that tolerance. The output of this pipeline is a list of row coordinates, each for a single line of text. For many applications, such as computer-assisted transcription, rather than exporting just the text line y coordinate, it is important to output an entire part of the original image. Thus, for each line we export a sub-image centered at the text line, with the width equal to the image width. The height is defined by the two minima of the y projected profile y_{pp}, one above and one below the y coordinate of the line (the maxima), and within a search interval equal to twice the height. See the output in the right side of Fig. 2.

7. ORIENTATION CORRECTION

The algorithm described above works well for images without strong skew. But in a more general case where the lines of the texts within images of scanned documents could have a certain amount of skew, we need to deskew the input images before applying our algorithm. This sub-section describes a simple but efficient pre-processing step that determines the text skew and orientation.

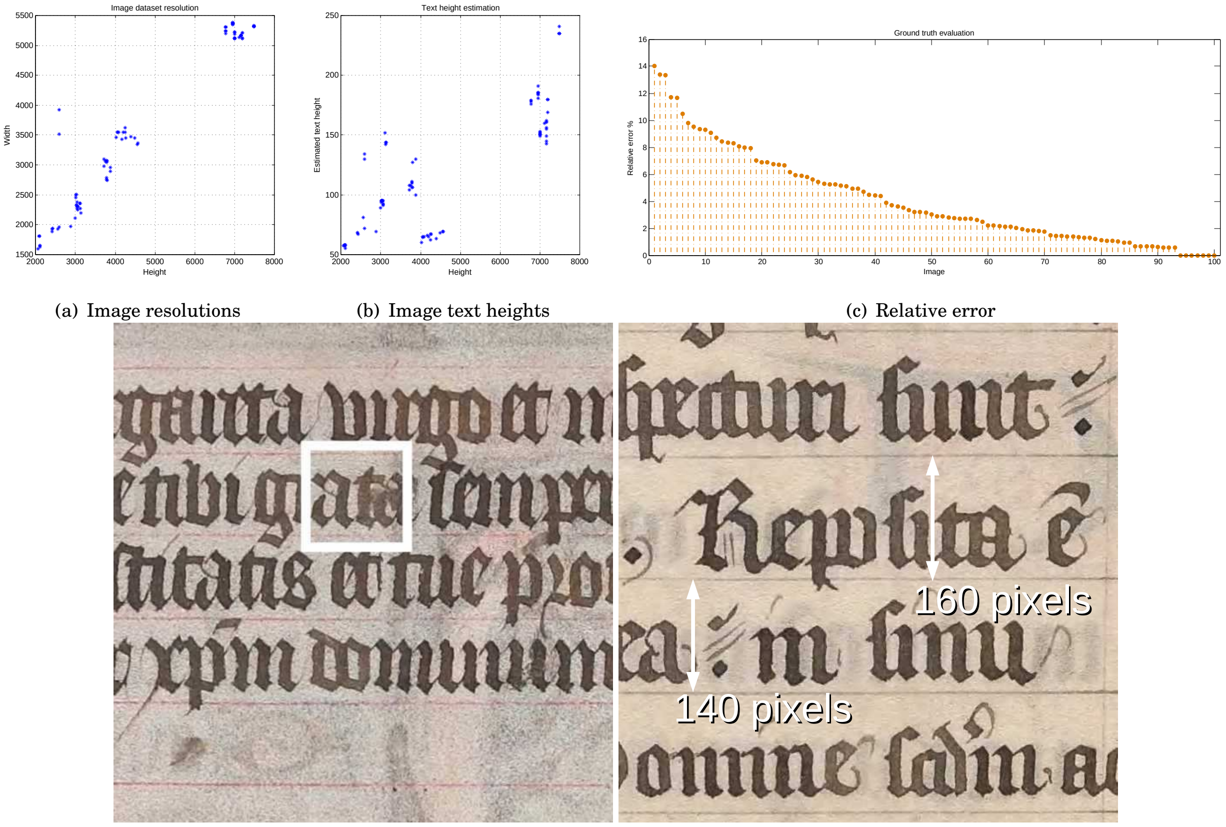
The main idea is based on the fact that the text within the test images has obvious vertical patterns with respect to one viewing direction (see Fig. 7(d) and 7(e)). Thus we can calculate the skew angle for a given input image by detecting the straight lines within it and looking into the statistics of the angles between these line segments and the x- or y-axis. More specifically, given a test image, we convert it into a binary image. Note that a number of image binarization algorithms have been proposed [Otsu 1975; Sauvola and Pietikinen 2000] and that we use the method by Otsu et al. [Otsu 1975] in this paper. After that, we utilize the recent line detection technique by Gioi et al. [Von Gioi et al. 2010] to detect all the line segments in the binary image. Assuming that (x_1, y_1) and (x_2, y_2) are the coordinates of the two endpoints of the i-th line segment and, without loss of generality, $y_2 \geq y_1$, we define the angle θ_i formed by this line segment and y-axis as

$$\theta_i = \begin{cases} \arccos\left(\dfrac{y_2 - y_1}{\|(x_2 - x_1, y_2 - y_1)\|}\right) & \text{if } x_1 \leq x_2 \\ -\arccos\left(\dfrac{y_2 - y_1}{\|(x_2 - x_1, y_2 - y_1)\|}\right) & \text{otherwise} \end{cases} \tag{7}$$

where $\|\cdot\|$ stand for the l^2-norm, and $\theta_i \in [-90°, 90]°$ is actually the angle between the vectors $(x_2 - x_1, y_2 - y_1)$ and (0, 1). Finally, we build a histogram of all θ_i with N bins and consider as the image skew angle θ the angle that corresponds to the peak of the histogram. Let $f(\alpha_i)$ be the frequency for the i-th histogram bin centered at α_i degrees. Then the skew angle θ is given by $\theta = \arg\max_{\alpha_i} f(\alpha_i)$. The sign of θ represents the direction of skew. That is, if $\theta < 0$, we need to rotate the image clockwise by $-\theta$ degrees to make the textlines parallel to the x-axis; otherwise, we make a counter-clockwise rotation by θ degrees.

8. RESULTS

We tested our algorithm on 34 Medieval manuscripts and 19 Arabic handwritten books (15552 pages). They are from the Yale University's Beinecke Rare Book and Manuscript Digital Library [Beinecke 2013b] (a set of scripts is available [Beinecke 2013a] to download a subset of the book database), the Oxford University's Bodleian Library, the Florence's Biblioteca Nazionale Centrale, the Walters Art Museum, the Admont's Stiftsbibliothek, the Köln's Erzbischöfliche Diözesan- und Dombibliothek, the Ripoll's Biblioteca Lambert Mata, the St. Gallen's Stiftsbibliothek, and the London's Wellcome Library. Compared to our previous work [Pintus et al. 2013], we add 32 new books with about 8600 more pages. In addition, we tested the proposed algorithm for segmentation of lines on a subset of these books (see Table IV), containing more than 80 thousands text lines. Those books are very different from each other, in terms of acquisition resolution (see Fig. 7(a)), level of conservation (e.g., noise, ink bleed-through and ageing), amount of figures and ornamentation, languages and writing styles. Our technique was implemented on Linux using C++ and the OpenCV library [OpenCV 2013]. Our benchmarks were executed on a PC with 8 Intel Core i7-3630QM CPU @ 2.40GHz processors, and 12GB RAM. By exploiting parallel execution of our code on the 8 processors, the average per-page computational time is about 3 minutes: ~ 6 seconds for text height, ~ 1 minutes for the coarse segmentation, ~ 1 minutes for SVM training, ~ 1 minutes for the feature classification, and ~ 8 seconds for text line extraction.



(a) Image resolutions (b) Image text heights (c) Relative error

(d) Image n.1 in (a) (e) Text height variability

Fig. 7. **Ground truth evaluation.** We consider a 100 image ground truth dataset. All image resolutions and ground truth text heights are respectively plotted in (a) and (b). The maximum relative error 8 is 14% (c), and corresponds to the square edge size t_E in (d). (e) shows one example of text variability across a single page. Manuscript images courtesy of the Yale University[BeineckeMS310 ; BeineckeMS10].

8.1 Text height extraction

Given an input page, it is very difficult to manually define a correct and unique text height, because it changes across the single page or even across a single line. Both when it is manually set [Mehri et al. 2013; Garz et al. 2012] and in our automatic estimation, the reliability of the text height value is inversely proportional to character size variability. We performed two different kinds of evaluations to understand if this computed value is below an acceptable error or not.

In the first case, we produced a ground truth dataset; we randomly took 100 images from all the datasets, with different resolutions (see Fig. 7(a)), text heights (see Fig. 7(b)) and types of manuscript, and we manually measured the text height for each of them. We then compared those values with the ones automatically computed by the algorithm. In Fig. 7(c) we plot the relative error of image i as

$$\epsilon_i = 100\frac{\left|t_E^i - \tilde{t}_E^i\right|}{\tilde{t}_E^i} \tag{8}$$

where t_E^i and $\tilde{t}_E^i$ are respectively the automatic and manual estimated values. In the plot we sort the errors in descending order. All the relative errors are under 15%, and in Fig. 7(d) we show the image corresponding to the highest relative error, in which we have drawn a square of edge size equal to t_E^i; the automatic estimated value well depicts the spatial periodicity of the analyzed text. It has a reasonable size for a general layout analysis approach [Mehri et al. 2013; Garz et al. 2012]. Further, in Fig. 7(e) we show how difficult it is for the user to choose a good height value; the spatial text period in two adjacent lines varies from 140 to 160 pixels, with a difference of about 15% between them. Thus, a 14% maximum relative error is an acceptable outcome.

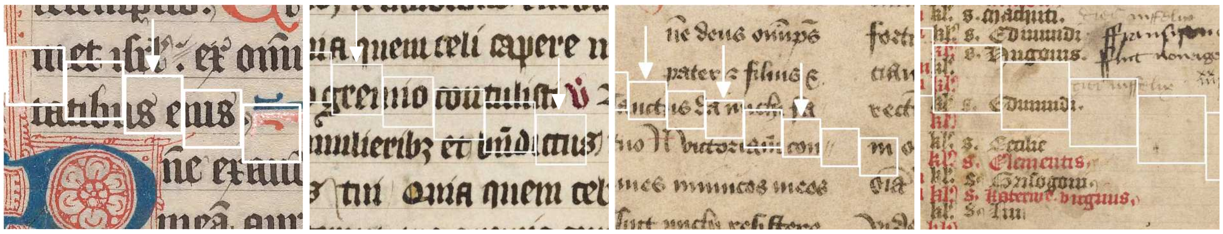

(a) (b) (c) (d)

Fig. 8. **Visual evaluation.** We present some check images, corresponding with acceptable (a)(b)(c) or unacceptable results (d). Manuscript images courtesy of the Yale University[BeineckeMS10] and of the Oxford University[BodleianMSBodley113 ; BodleianMSBodley850].

However, this evaluation is only practical for a small subset of images. We would like to check all the thousands images in the studied books. This can only be done visually in a computer assisted framework. Hence, for each image, after computing the text height value t_E, the algorithm draws a pattern of nine squares with edge equal to t_E. The original image with these overlapping squares helps the user to quickly estimate if the analysis result is visually reasonable. To understand the reliability of such evaluation, in Fig. 8 we show details of some checked images, corresponding with both acceptable and unacceptable results. We highlight with arrows the squares that were particularly helpful to us in marking the outcome as a good one. Table I shows a high rate of good estimations; the majority of the books are over 95% accurate, with some even achieving 100% accuracy. The overall dataset of 53 books (15552 pages) is over 98% accurate.

Typical pages of illuminated manuscripts are shown in Fig. 9. They contain text in two different colors, capital letters of different types and sizes, the parchment background, other figures inside the text and ornaments. The images could also contain the dark acquisition background, and other visible parts of the book. We present both the original images of the page and one or more highlighted parts, with a square of edge size equal to the estimated text height t_E. This helps demonstrate the conditions of the whole analysis domain, and to visually appreciate the quality of the output. Although the result

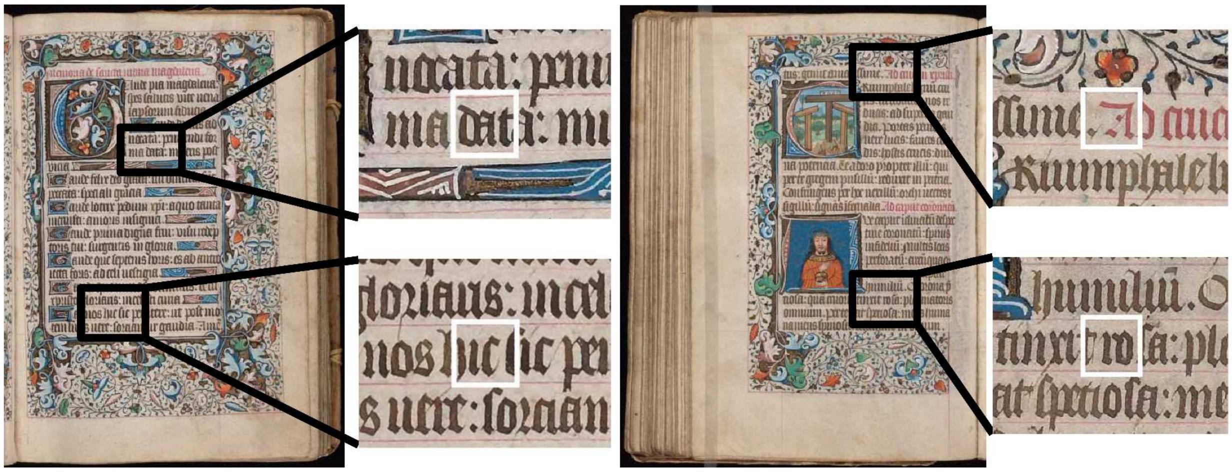
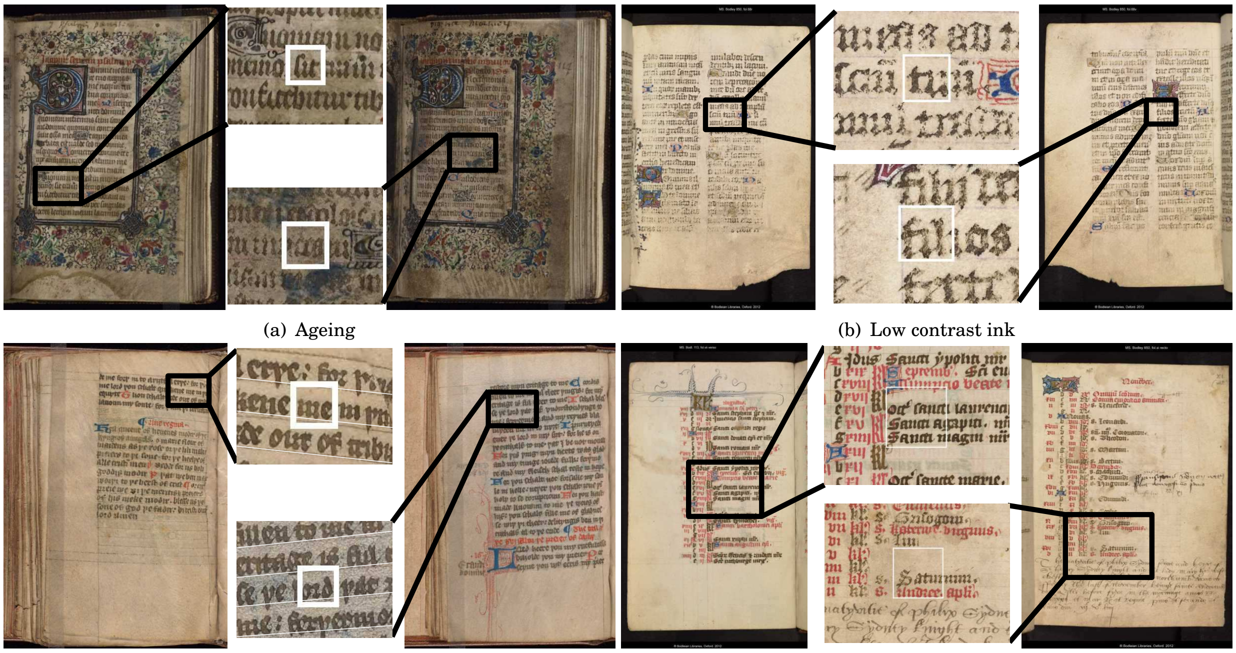


Fig. 9. **Illuminated manuscript pages.** We present the original image of the page and two highlighted parts; the squares have edge lengths equal to the automatically estimated text height t_E. Manuscript images courtesy of the Yale University[BeineckeMS310].



(a) Ageing

(b) Low contrast ink

(c) Skew text

(d) Failures

Fig. 10. **Challenging samples and failures.** These pages are affected by the following imperfections: (a) strong ageing; (b) low contrast ink; (c) skewed text. In (d) the algorithm fails due to a lack of a predominant spatial frequency. Manuscript images courtesy of the Yale University [BeineckeMS109 ; BeineckeMS360] and the Oxford Library [BodleianMSBodley850].

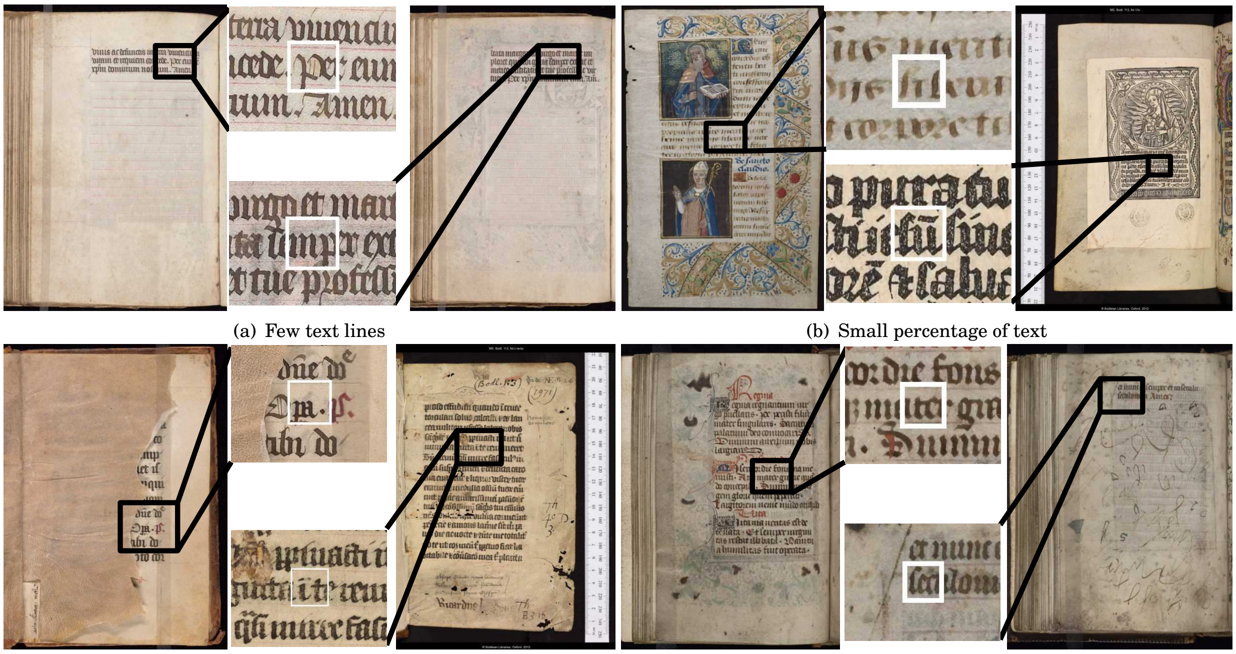
is good, the pages in Fig. 9 are not so challenging, since they are very well preserved and do not contain any kind of noise. In Fig. 10 and Fig. 11, we present most of the problems that arise when dealing with very old handwritten manuscripts. The two pages in Fig. 10(a) are affected by significant ageing and very bad preservation conditions; the result shows how the proposed frequency-based descriptor is able to extract the main image directions even at the presence of a very noisy signal. It is also robust to low constrast signals, as shown in Fig. 10(b), where the ageing makes the ink almost disappear. Due to the value of these rare books, the acquisition setup is carefully controlled, both to preserve their integrity, and to produce the best possible digital images. However, some texts can come out non-horizontal, if the text is skewed relative to the page edges. In our tests we never needed to use an automatic skew correction, and Fig. 10(c) proves how the proposed technique is able to properly deal with skewed texts. The multi-scale framework is convenient when we need to cope with other extreme but common situations, such as a low number of text lines (Fig. 11(a)) or a small percentage of text in a page with a lot of figures and other non-text elements (Fig. 11(b)). Two extreme cases are presented in Fig. 11(c); in one case, only a small part of the text is visible, and, in the other, the page is very damaged and contains a lot of comments written in different styles. We also demonstrate how our approach is robust to the well known problem of ink bleed-through, which makes the writing from the verso appear on the recto (Fig. 11(d)). The page on the right in Fig. 11(d) is affected by bleed-through, it contains very few text lines, a lot of noise and other handwritten signs. Since our method aims at finding the most predominant spatial periodicity in the page, we have seen that it fails when there are some concurrent high amplitude frequencies, or in the presence of sparsely inscribed pages. This occurs when the text is not organized in a regular manner, or, in other words, when the inter-line spacing has high variability. On the other hand, if the quasi-horizontal text constraint is met, locally skewed text won't affect the text height estimation, due to the redundant, global information from the rest of the image. The failures (bad images) in table I are always similar to those in Fig. 10(d); on the left the bad estimated text height value clearly depends on the groups of three text lines, while the case of the right contains both a non-regular text line pattern and an additional comment part in the bottom, written (perhaps by a different author) with a completely different style. The worst result in table I is due to a book (*BodleianMSBodley920*) that contains a large amount of pages with non-regular text layout.

We also tested the proposed text height estimation to non-Latin writing styles, in order to prove its applicability in a more general framework. In Fig. 12 we show handwritten documents in three different languages, i.e., Andalusian, Kufi and Chinese. In the latter case, Chinese is vertically oriented, so the images should be tagged accordingly; the algorithm will simply rotate them before computing text height value. Moreover, table I shows the algorithm performance on 19 arabic manuscripts. These examples prove our method is independent of the type of language.

Although the automatic text height estimation is just the first important step to building a completely automatic layout analysis framework, this simple output can lead to some very useful results. It turns out that the text size measured across all the pages of a single book is somehow consistent, while the text height estimation for pages without any text is random. By exploiting the text height and the color statistical distribution (average and variance) across the same book, we can distinguish between pages that contain text and pages that contain only figures. In table II we show the precision/recall results after applying this segmentation to books having pages with only figures. The true positive are the pages well segmented that contain only figures, while the false positive/negative are bad segmented pages that respectively do not/do contain only figures. In our experiments the recall value is equal to 1 because we do not have any false negatives. Another possible application could be a tool that gives scholars an ordered list of pages based on the computed statistical distribution; i.e., users can analyze a book by first sorting its pages according to the level of image or text content.

8.2 Text line segmentation

With the same motivation as for the text height, we perform two types of evaluation. First, we randomly took 55 pages (i.e., 5 for each book in table IV), which contain 1019 lines, and we manually


(a) Few text lines

(b) Small percentage of text

(c) Significant damage

(d) Ink bleed-through

Fig. 11. **Challenging samples.** These pages are affected by the following imperfections: (a) few text lines; (b) a small quantity of text; (c) significant damage; (d) ink bleed-through. Manuscript images courtesy of the Yale University [BeineckeMS310 ; BeineckeMS109] and the Oxford Library [BodleianMSBodley113].

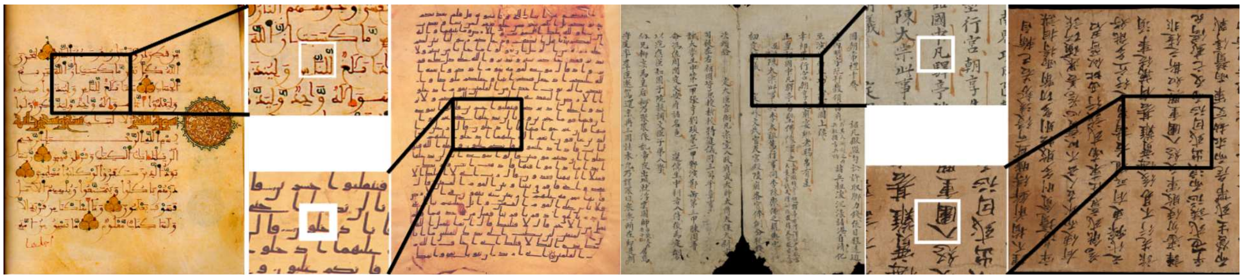


Fig. 12. **Non-latin manuscript pages.** We present the original image of the page and one highlighted part in the case of non-latin page. From left to right: andalusian [AndalusQuran], kufi [KufiQuran] and two chinese writing [ChineseManuscript b; a]. The squares have edge lengths equal to the automatically estimated text height t_E.

segment the lines of text. For each line, we perform a per-pixel comparison between the manually extracted sub-image and the one automatically computed by our algorithm. For each pixel, we evaluate (and mark it accordingly) whether it is a *true-positive* (TP), *false-positive* (FP) or *false-negative* (FN). A *true-positive* is a correct segmented pixel, a *false-positive* is a pixel that was extracted from the page but did not belongs to an actual line (also known as a *false alarm*), while a *false-negative* is a pixel that is not extracted from the page but it belongs to a line (also known as a *miss*). We then compute the

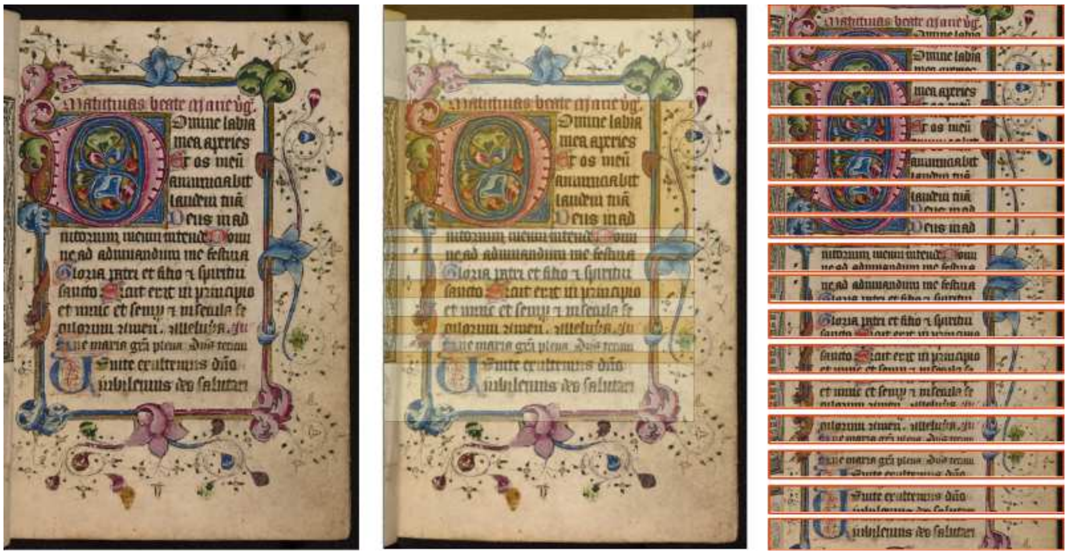


Fig. 13. **Our technique vs T-pen.** A comparison between the proposed automatic text line segmentation method and the one employed in T-pen [T-pen 2013], a broadly used web service for computer assisted transcription. From left to right: a page from a manuscript, line segmentation of T-pen, our line extraction. Manuscript images courtesy of the Oxford University[BodleianMSBodley113].

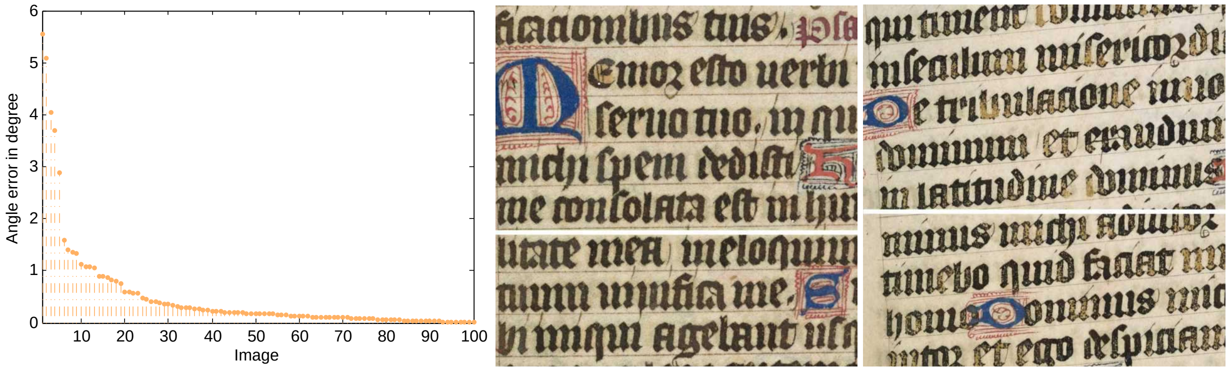
corresponding *Precision* and *Recall* values as:

$$\begin{cases} Precision = \frac{TP}{TP+FP} \\ Recall = \frac{TP}{TP+FN} \end{cases} \tag{9}$$

In this case, the *Precision* means the probability that a randomly selected retrieved pixel is relevant (TP), while the *Recall* is the probability that a randomly selected pixel belonging to a line of text in the database will be retrieved in a search. Since a manual segmentation is not an exact *ground truth*, but has some degree of variability, we perform two different manual segmentations $M1$ and $M2$, done by different people. In table III we compare all the three segmentations (i.e., one automatic A, and the two manual ones) in a combinatorial manner. For each comparison we take a reference set of pixels as a *ground truth*, and another set as a computed data. Our conservative line extraction has a high *Recall* value ($> 96\%$), but, as expected, lack some *Precision*. However, the comparison between $M1$ and $M2$ shows the high variability of manual classification, and the maybe ill-posed problem of defining a real, unique *ground truth*.

In order to provide a more extensive evaluation, we launch the algorithm on a big amount of images (11 books with 80963 lines) in our database, and we perform a visual analysis across the set of subimages, each corresponding to a segmented line of text. Similarly, for each line we evaluate whether it is a TP, FP or FN. In table IV for each book we report the total number of lines it contains, the TP, FP and FN, and the resulting *Precision* and *Recall* values. As we can see, the modified, automatic and parameter-free approach based on projection profiles and text height estimation produces *Precision* and *Recall* values that are always above 93%.

One of the most common use of segmented lines of text is to assist users in transcription. As stated above, although a deep test of our approach against all other research on text line segmentation is out of the scope of this paper, we want to report a practical comparison between the output produced by our method and one from a widely used web service for transcription [T-pen 2013]. In Fig. 13 we see how our technique works better if the manuscript contains a lot of illumination, ornamentation, figures and capital letters.



(a) (b) (c)

Fig. 14. **Text orientation correction.** (a) Angle error. Close-up of an image with angle error of 1.58 degrees (b) and 5.54 degrees (c). Manuscript images courtesy of the Oxford University[BodleianMSBodley113].

8.3 Text orientation correction

To evaluate the performance of the method for text orientation correction, we arbitrarily rotated the 100 randomly selected images and applied the method to the rotated images to obtain the skew angles θ. In this experiment, we fix the number of histogram bins at $N = 360$. Fig. 14 (a) shows the absolute angle error/difference between the computed skew angles θ and their corresponding groundtruth rotation angles. From this figure, we see that our orientation correction method is able to find the skew angle at a satisfactory rate, that is, with an angle error of less than 1.5 degrees, up to 94% of the time. After comparing the white horizontal reference line in Fig. 14 (b) and the text orientation, it is clear that the error of 1.5 degrees is trivial and thus absolutely acceptable for practical applications. Fig. 14 (c) shows a close-up of the image that corresponds to the highest angle error of 5.54 degrees. Although the error is 5.54 degrees, our method actually outputs the expected skew angle because we can, upon close inspection, observe that the strokes of the texts are approximately perpendicular to the white horizontal reference line. In addition, it is worth mentioning here that the proposed text height estimation algorithm can tolerate a skew of up to 6 degrees (see Fig. 10 (c)).

9. CONCLUSION

We have presented a method to perform automatic text height estimation, with no manual intervention and user defined parameters, and an automatic framework to extract lines of text in old handwritten books. We have tested our algorithms on a large heterogeneous corpus of books; the algorithms proved to be very robust and reliable for very noisy and damaged manuscripts, with different writing styles, languages, text sizes, image resolutions, levels of conservation, and for those affected by numerous uncontrollable factors, such as holes, spots, ink bleed-through, ornamentation, background noise, and touching text lines. Future work will investigate and try to deal with some limitations of the current methods, such as varying text height in the same page and non-regular text structures. We will study a more efficient multi-scale approach, e.g., with an adaptive domain refinement based on the local normalized autocorrelation output at each node of the multi-level hierarchy. Further, we will investigate other local image descriptors, as dense sift [Wang et al. 2010] or DAISY [Tola et al. 2010], in order to understand how those behave with handwritten text and very noisy data. Due to the intrinsic parallel nature of our analysis, a GPU-based implementation is straightforward, and would make it more suitable for processing larger databases. We will employ a text block segmentation framework in order to extract text lines in a more general scenario, e.g., two column layout books, and we will exploit the redundancy across an entire book to increase the *Precision* and *Recall* performances, and the robustness of the proposed per-page basis algorithm. Finally we will extensively evaluate our text line extraction compared to the state-of-the-art approaches, and using both our database and publicy available benchmarks [Antonacopoulos et al. 2009; Antonacopoulos et al. 2013; Stamatopoulos et al. 2013; Kleber et al. 2013].

Acknowledgments. This work was supported by the Digitally Enabled Scholarship with Medieval Manuscripts (DESMM) project funded by the Mellon Foundation (http://ydc2.yale.edu/). The authors would also like to thank Chang Liu for her assistance in preparing the groundtruth data used in this paper. All the manuscript images in the paper were retrieved from an International Image Interoperability Framework-compatible image server hosted by Yale University on DATE. Images were created with funding from the Andrew W. Mellon Foundation and are provided under the Creative Commons CC BY license.

REFERENCES

ALMEIDA, M. S. AND ALMEIDA, L. B. 2012. Nonlinear separation of show-through image mixtures using a physical model trained with ica. *Signal Processing 92,* 4, 872–884.

ANDALUSQURAN. http://en.wikipedia.org/wiki/Criticism_of_Islam.

ANTONACOPOULOS, A., CLAUSNER, C., PAPADOPOULOS, C., AND PLETSCHACHER, S. 2013. Icdar 2013 competition on historical newspaper layout analysis (hnla 2013). In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. 1454–1458.

ANTONACOPOULOS, A. AND KARATZAS, D. 2004. Document image analysis for world war ii personal records. In *Proc. Document Image Analysis for Libraries, 2004.* 336–341.

ANTONACOPOULOS, A., PLETSCHACHER, S., BRIDSON, D., AND PAPADOPOULOS, C. 2009. Icdar 2009 page segmentation competition. In *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*. 1370–1374.

BAR-YOSEF, I., HAGBI, N., KEDEM, K., AND DINSTEIN, I. 2009. Line segmentation for degraded handwritten historical documents. In *ICDAR*. IEEE, 1161–1165.

BEINECKE. 2013a. 21 book database - download scripts - http://hdl.handle.net/10079/cz8w9v8.

BEINECKE. 2013b. Beinecke rare book and manuscript library - http://beinecke.library.yale.edu/.

BEINECKEMS10. Beinecke rare book and manuscript library - yale university.

BEINECKEMS109. Beinecke rare book and manuscript library - yale university.

BEINECKEMS310. Beinecke rare book and manuscript library - yale university.

BEINECKEMS360. Beinecke rare book and manuscript library - yale university.

BODLEIANMSBODLEY113. Bodleian library - oxford university.

BODLEIANMSBODLEY850. Bodleian library - oxford university.

BULACU, M., VAN KOERT, R., SCHOMAKER, L., VAN DER ZANT, T., ET AL. 2007. Layout analysis of handwritten historical documents for searching the archive of the cabinet of the dutch queen. In *ICDAR*. 357–361.

CHINESEMANUSCRIPT. British library - http://en.wikipedia.org/wiki/Dunhuang_Go_Manual.

CHINESEMANUSCRIPT. http://leminhkhai.files.wordpress.com/2013/08/manuscript.jpg.

DIEM, M. AND SABLATNIG, R. 2010. Recognizing characters of ancient manuscripts. In *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 753106–753106.

DIGITALHERITAGE. International congress 2013 - http://www.digitalheritage2013.org/.

EGLIN, V., BRES, S., AND RIVERO, C. 2007. Hermite and gabor transforms for noise reduction and handwriting classification in ancient manuscripts. *IJDAR 9,* 2-4, 101–122.

GARZ, A., DIEM, M., AND SABLATNIG, R. 2010. Local descriptors for document layout analysis. In *Advances in Visual Computing*. Springer, 29–38.

GARZ, A., FISCHER, A., SABLATNIG, R., AND BUNKE, H. 2012. Binarization-free text line segmentation for historical documents based on interest point clustering. In *DAS 2012*. 95–99.

GRANA, C., BORGHESANI, D., AND CUCCHIARA, R. 2009. Picture extraction from digitized historical manuscripts. In *Proceedings of the ACM International Conference on Image and Video Retrieval*. ACM, 22.

JAIN, A. K. AND NAMBOODIRI, A. M. 2003. Indexing and retrieval of on-line handwritten documents. In *ICDAR*. 655.

JINDAL, S. AND LEHAL, G. S. 2012. Line segmentation of handwritten gurmukhi manuscripts. In *DAR*. ACM, 74–78.

JOURNET, N., MULLOT, R., RAMEL, J.-Y., AND EGLIN, V. 2005. Ancient printed documents indexation: a new approach. In *Pattern Recognition and Data Mining*. Springer, 580–589.

JOURNET, N., RAMEL, J.-Y., MULLOT, R., AND EGLIN, V. 2008. Document image characterization using a multiresolution analysis of the texture: application to old documents. *IJDAR 11,* 1, 9–18.

JOUTEL, G., EGLIN, V., AND EMPTOZ, H. 2008. A complete pyramidal geometrical scheme for text based image description and retrieval. In *Image and Signal Processing*. Springer, 471–480.

KHEDEKAR, S., RAMANAPRASAD, V., SETLUR, S., AND GOVINDARAJU, V. 2003. Text-image separation in devanagari documents. In *ICDAR*. Vol. 2.

KLEBER, F., FIEL, S., DIEM, M., AND SABLATNIG, R. 2013. Cvl-database: An off-line database for writer retrieval, writer identification and word spotting. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, 560–564.

KOO, H. I. AND CHO, N. I. 2010. State estimation in a document image and its application in text block identification and text line extraction. In *Computer Vision–ECCV 2010*. Springer, 421–434.

KOO, H. I. AND CHO, N. I. 2012. Text-line extraction in handwritten chinese documents based on an energy minimization framework. *Image Processing, IEEE Transactions on 21,* 3, 1169–1175.

KUFIQURAN. http://free-minds.org/forum/index.php?topic=9602163.0.

LEMAITRE, A., CAMILLERAPP, J., AND COÜASNON, B. 2008. Multiresolution cooperation makes easier document structure recognition. *IJDAR 11,* 2, 97–109.

LEYDIER, Y., LEBOURGEOIS, F., AND EMPTOZ, H. 2007. Text search for medieval manuscript images. *Pattern Recognition 40,* 12, 3552–3567.

LIKFORMAN-SULEM, L., ZAHOUR, A., AND TACONET, B. 2007. Text line segmentation of historical documents: a survey. *IJDAR 9,* 2-4, 123–138.

LOULOUDIS, G., GATOS, B., PRATIKAKIS, I., AND HALATSIS, C. 2008. Text line detection in handwritten documents. *Pattern Recognition 41,* 12, 3758–3772.

LOWE, D. G. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision 60,* 2, 91–110.

MEHRI, M., GOMEZ-KRÄMER, P., HÉROUX, P., AND MULLOT, R. 2013. Old document image segmentation using the autocorrelation function and multiresolution analysis. In *IS&T/SPIE Electronic Imaging*.

MINETTO, R., THOME, N., CORD, M., LEITE, N. J., AND STOLFI, J. 2012. T-hog: An effective gradient-based descriptor for single line text regions. *Pattern Recognition*.

NAGY, G. 2000. Twenty years of document image analysis in pami. *PAMI 22,* 1, 38–62.

OPENCV. 2013. Opencv - open source computer vision library.

OTSU, N. 1975. A threshold selection method from gray-level histograms. *Automatica 11,* 285-296, 23–27.

PAPANDREOU, A., GATOS, B., LOULOUDIS, G., AND STAMATOPOULOS, N. 2013. Icdar 2013 document image skew estimation contest (disec 2013). In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, 1444–1448.

PINTUS, R., YANG, Y., AND RUSHMEIER, H. 2013. Athena: Automatic text height extraction for the analysis of old handwritten manuscripts. *Digital Heritage 2013 6,* 4.

QIAO, Y.-L., LU, Z.-M., SONG, C.-Y., AND SUN, S.-H. 2006. Document image segmentation using gabor wavelet and kernel-based methods. In *ISSCAA*. 5–pp.

RATZLAFF, E. H. 2000. Inter-line distance estimation and text line extraction for unconstrained online handwriting. In *Workshop on Frontiers in Handwriting Recognition*. 33–42.

SABHARWAL, C. L. AND SUBRAMANYA, S. 2001. Indexing image databases using wavelet and discrete fourier transform. In *Proceedings of the 2001 ACM symposium on Applied computing*. ACM, 434–439.

SAUVOLA, J. AND PIETIKINEN, M. 2000. Adaptive document image binarization. *Pattern Recognition 33,* 2, 225 – 236.

SHAPIRO, V., GLUHCHEV, G., AND SGUREV, V. 1993. Handwritten document image segmentation and analysis. *Pattern Recognition Letters 14,* 1, 71–78.

SHARMA, N., PAL, U., AND BLUMENSTEIN, M. 2012. Recent advances in video based document processing: a review. In *DAS*. 63–68.

SHI, Z., SETLUR, S., AND GOVINDARAJU, V. 2009. A steerable directional local profile technique for extraction of handwritten arabic text lines. In *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*. IEEE, 176–180.

STAMATOPOULOS, N., GATOS, B., LOULOUDIS, G., PAL, U., AND ALAEI, A. 2013. Icdar 2013 handwriting segmentation contest. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. 1402–1406.

T-PEN. 2013. T-pen - transcription for paleographical and editorial notation.

TOLA, E., LEPETIT, V., AND FUA, P. 2010. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 32,* 5, 815–830.

VON GIOI, R. G., JAKUBOWICZ, J., MOREL, J.-M., AND RANDALL, G. 2010. Lsd: A fast line segment detector with a false detection control. *PAMI, IEEE Transactions 32,* 4, 722–732.

WANG, J.-G., LI, J., LEE, C. Y., AND YAU, W.-Y. 2010. Dense sift and gabor descriptors-based face representation with applications to gender recognition. In *Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on*. IEEE, 1860–1864.

WANG, Y., PHILLIPS, I. T., AND HARALICK, R. M. 2002. A study on the document zone content classification problem. In *DAS*. 212–223.

YALNIZ, I. Z. AND MANMATHA, R. 2012. An efficient framework for searching text in noisy document images. In *DAS*. 48–52.

ZAHOUR, A., TACONET, B., MERCY, P., AND RAMDANE, S. 2001. Arabic hand-written text-line extraction. In *ICDAR*. IEEE, 281–285.

Table I. Text height estimation statistics.

Book Name	Avg. Resolution WxH	Pages	Non-text	Text	Good	Bad	Time
BeineckeMS310	2886 x 3794	309	32	277	266 (96%)	11 (4%)	12m (2sec/pg)
BeineckeMS10	2324 x 3127	187	13	174	174 (100%)	0 (0%)	4m (1sec/pg)
BeineckeMS109	1822 x 2416	270	19	251	251 (100%)	0 (0%)	4m (1sec/pg)
BeineckeMS360	1654 x 2083	382	11	371	371 (100%)	0 (0%)	3m (0.5sec/pg)
BeineckeMS748	2313 x 3232	8	0	8	8 (100%)	0 (0%)	11s (1sec/pg)
BeineckeMS525	2053 x 2855	46	4	42	42 (100%)	0 (0%)	1m (1sec/pg)
BodleianMSBodley113	5329 x 7487	315	12	303	292 (96%)	11 (4%)	75m (14sec/pg)
BodleianMSBodley850	5370 x 6959	246	16	230	217 (94%)	13 (6%)	41m (10sec/pg)
BodleianMSDouce18	5167 x 7155	534	17	517	505 (98%)	12 (2%)	2h27m (14sec/pg)
BodleianMSGoughLiturg.3	5278 x 6786	257	36	221	219 (99%)	2 (1%)	1h (14sec/pg)
BodleianMSLaudMisc.204	5170 x 7013	286	36	250	246 (98%)	4 (2%)	1h38m (21sec/pg)
BodleianMSliturg.e.17	5237 x 7201	224	13	211	209 (99%)	2 (1%)	36m (10sec/pg)
MarstonMS22	3814 x 2574	121	5	116	116 (100%)	0 (0%)	4m (2sec/pg)
Osborna44	2336 x 3025	483	13	470	463 (98%)	7 (2%)	12m (2sec/pg)
Osbornfa1	3487 x 4240	400	3	397	396 (99%)	1 (1%)	28m (4sec/pg)
Walters34	2050 x 3139	658	78	580	574 (99%)	6 (1%)	12m (1sec/pg)
Walters102	2291 x 3359	222	14	208	208 (100%)	0 (0%)	6m (2sec/pg)
Admont43	2882 x 4347	358	0	358	358 (100%)	0 (0%)	17m (3sec/pg)
Admont23	2815 x 4286	591	0	591	591 (100%)	0 (0%)	27m (3sec/pg)
CologneErzbisch127Ka	3480 x 4491	625	10	615	614 (99%)	1 (1%)	16m (2sec/pg)
CologneErzbisch128Kb	3072 x 3840	400	1	399	399 (100%)	0 (1%)	5m (1sec/pg)
BodleianMSAuctDinf.2.11	4945 x 7126	544	52	492	456 (93%)	36 (7%)	1h46m (12sec/pg)
BodleianMSBodley716	5192 x 7174	524	23	501	489 (98%)	12 (2%)	2h41m (18sec/pg)
BodleianMSBodley861	5324 x 7000	352	31	321	321 (100%)	0 (0%)	1h24m (14sec/pg)
BodleianMSBodley920	5070 x 6940	209	12	197	152 (77%)	45 (23%)	56m (16sec/pg)
BodleianMSDouce231	4317 x 6201	238	22	216	190 (88%)	26 (12%)	47m (12sec/pg)
BodleianMSGoughLiturg.19	3684 x 4573	236	15	221	213 (96%)	8 (4%)	19m (5sec/pg)
BodleianMSLatLiturg.f.21	5249 x 6046	196	16	180	180 (100%)	0 (0%)	34m (10sec/pg)
BodleianMSLatliturg.f.2	4925 x 5764	356	47	309	307 (99%)	2 (1%)	45m (8sec/pg)
BodleianMSLaudLat.4	5117 x 7456	568	22	546	546 (100%)	0 (0%)	2h27m (16sec/pg)
BodleianMSLaudMisc.188	5075 x 6831	620	12	608	598 (98%)	10 (2%)	2h35m (15sec/pg)
FlorenceBibNazCen402Fd	4032 x 2908	175	7	168	168 (100%)	0 (0%)	5m (2sec/pg)
Ripoll078	1944 x 2592	357	0	357	355 (99%)	2 (1%)	2m (1sec/pg)
SanktGallen673Sg	3328 x 4992	248	4	244	244 (100%)	0 (0%)	5m (1sec/pg)
WMS_Arabic_1	975 x 773	537	11	526	526 (100%)	0 (0%)	50s (<1sec/pg)
WMS_Arabic_6	975 x 732	101	27	74	73 (99%)	1 (1%)	15s (<1sec/pg)
WMS_Arabic_8	975 x 772	64	11	53	53 (100%)	0 (0%)	4s (<1sec/pg)
WMS_Arabic_18	975 x 732	349	12	337	337 (100%)	0 (0%)	34s (<1sec/pg)
WMS_Arabic_20	975 x 716	310	19	291	291 (100%)	0 (0%)	47s (<1sec/pg)
WMS_Arabic_22	986 x 686	449	18	431	430 (99%)	1 (1%)	47s (<1sec/pg)
WMS_Arabic_24	975 x 635	54	14	40	40 (100%)	0 (0%)	3s (<1sec/pg)
WMS_Arabic_26	979 x 738	116	15	101	99 (98%)	2 (2%)	28s (<1sec/pg)
WMS_Arabic_29	981 x 760	231	11	220	220 (100%)	0 (0%)	51s (<1sec/pg)
WMS_Arabic_31	984 x 773	87	17	70	68 (97%)	2 (3%)	9s (<1sec/pg)
WMS_Arabic_34	986 x 693	49	12	37	37 (100%)	0 (0%)	6s (<1sec/pg)
WMS_Arabic_35	986 x 793	235	9	226	226 (100%)	0 (0%)	51s (<1sec/pg)
WMS_Arabic_46	985 x 744	214	10	204	204 (100%)	0 (0%)	51s (<1sec/pg)
WMS_Arabic_49	978 x 1564	370	19	351	351 (100%)	0 (0%)	2m (<1sec/pg)
WMS_Arabic_62	986 x 743	142	11	131	131 (100%)	0 (0%)	36s (<1sec/pg)
WMS_Arabic_66	986 x 807	197	9	188	187 (99%)	1 (1%)	52s (<1sec/pg)
WMS_Arabic_68	986 x 737	36	15	21	20 (95%)	1 (5%)	5s (<1sec/pg)
WMS_Arabic_72	989 x 745	141	7	134	133 (99%)	1 (1%)	37s (<1sec/pg)
WMS_Arabic_80	987 x 744	325	12	313	313 (100%)	0 (0%)	52s (<1sec/pg)
# books - 53		15552	855	14697	14477 (>98%)	220 (<2%)	24h46m (6sec/pg)

Table II. Precision and Recall values in the retrieval of those pages that contain only figures.

Book Name	# Figure Pages	Precision	Recall
BeineckeMS310	23	0.92	1.0
BodleianMSGoughLiturg.3	5	0.45	1.0
BodleianMSLaudMisc.204	16	0.88	1.0
Walters34	26	0.89	1.0

Table III. Text line quantitative evaluation. Classification error of an automatic (A) and two manual ($M1$, $M2$) segmentations.

Reference	Data	True-positive	False-positive	False-negative	Precision	Recall
M1	A	$\sim$ 356Mpixels	$\sim$ 238Mpixels	$\sim$ 9Mpixels	59.95%	97.58%
M2	A	$\sim$ 335Mpixels	$\sim$ 259Mpixels	$\sim$ 11Mpixels	56.46%	96.80%
M1	M2	$\sim$ 328Mpixels	$\sim$ 18Mpixels	$\sim$ 35Mpixels	94.88%	90.36%

Table IV. Text line segmentation statistics.

Book Name	Lines	True-positive	False-positive	False-negative	Precision	Recall
BeineckeMS109	4616	4511	62	105	98.64%	97.73%
BeineckeMS310	5299	4967	62	332	98.77%	93.73%
Admont23	18855	18528	376	327	98.01%	98.27%
BodleianMSAuctDinf.2.11	7576	7038	273	538	96.27%	92.90%
BeineckeMS10	2086	2084	0	2	100.00%	99.90%
BeineckeMS360	8186	8041	113	145	98.61%	98.23%
MarstonMS22	1512	1484	10	28	99.33%	98.15%
Walters34	8195	7800	6	395	99.92%	95.18%
Walters102	3911	3870	42	41	98.93%	98.95%
Osborna44	9277	8384	3	893	99.96%	90.37%
Admont43	11450	11268	199	182	98.26%	98.41%
# books - 11	80963	77975	1146	2988	98.55%	96.31%