Text Baseline Detection, a single page trained system

Moisés Pastor
Email address: mpastorg@prhlt.upv.es

Pattern Recognition and Human Language Technologies research centre
Universitat Politècnica de València
46022 València Spain

Abstract

Nowadays, there are a lot of page images available and the scanning process is quite well resolved and can be done industrially. On the other hand, HTR systems can only deal with single text line images. Segmenting pages into single text line images is a very expensive process which has traditionally been done manually. This is a bottleneck which is holding back any massive industrial document processing. A baseline detection method will be presented here[1]. The initial problem is reformulated as a clustering problem over a set of interest points. Its design aim is to be fast and to resist the noise artifacts that usually appear in historical manuscripts: variable interline spacing, the overlapping and touching of words in adjacent lines, humidity spots, etc. Results show that this system can be used to massively detect where the text lines are in pages. Highlight: This system reached second place in the ICDAR 2017 Competition on Baseline Detection (see Table 1).

1. Introduction

Over the past decades, numerous libraries and archives have made a great effort to digitize their collections. As a consequence, a huge amount of historical handwritten document images have been published in online digital libraries.

[1] freely available at https://github.com/moisesPastor/baseLinePage

	Precision	Recall	F-Measure
DMRZ	0.973	0.970	0.971
UPVLC	0.937	0.855	0.894
BYU	0.878	0.907	0.892
IRISA	0.883	0.877	0.880
LITIS	0.780	0.836	0.807

Table 1: ICDAR 2017 Competition on Baseline Detection. Results for the cBAD test set track-a [1]

This allows for the protection of the originals while sharing access to the information within them. Nevertheless, methods need to be developed to make these documents searchable, thereby making them useful for historians and other researchers. Transcribing such a large amount of documents manually is both time and cost prohibitive. Currently, transcriptions are obtained automatically with a posteriori human revision or by using computer-assisted engines [2, 3, 4] where the user collaborates interactively with the system to get the perfect transcription. Word spotting systems have also been developed to search through collections that have not been previously transcribed [5, 6, 7]. So far, automatic (or assisted) handwritten text recognizers need to be fitted with segmented text line images. Nevertheless, there are a huge amount of scanned manuscript page images available. The process of segmenting text page images into text line images is a bottleneck which is holding back any massive industrial document processing.

At this point, two related problems can be defined: *baseline detection* and *text line extraction*. *Baseline detection* is a substantial process in document image analysis that can be used not only to segment page images into line images but also for many other document processing steps such as skew, slant, and slope correction; text height normalization [8, 9]; feature extraction; or the rectification of geometric distortions [10, 11]. A baseline is a fictitious line which follows and joins the lower part of the character bodies in a text line [12]

and below which descenders extend. These baselines are not expensive to obtain manually or semi-automatically and are used to know where the lines are. In the work of V. Romero et al. [13], baselines were used to extract the text lines images and their performance was compared to that of manually segmented lines. Text line extraction consists of defining a set of page image regions covering all page lines and containing a single text line per region [14, 15].

The baseline detection and *text line extraction* processes are not easy tasks to complete in manuscript texts when compared with printed texts. Manuscript texts present some challenging problems including different skews, variable interline spacing, the overlapping and touching of words between adjacent lines, etc. These problems are even worse in the case of historical documents due to the degradation problems they suffer as smear, significant background variations, uneven illumination, a lack of contrast, humidity spots, bleed-through, layout inconsistencies, decorative entities, etc. Segmenting page images into lines images is a very expensive process and is another bottleneck which is holding back any massive industrial document processing. The aim of the present work is to contribute to overcoming this bottleneck.

The main contribution of this work consists in presenting a state-of-the-art baseline detection and extraction system which is noise-resistant and which only requires one page to be trained and which requires no special hardware in order to work. Also, we bring the whole software source code to the community via GitHub.

This paper is organized as follows: Section 2 provides a brief summary of related work. The baseline detection method proposed is presented in Section 3. In Section 4, a line extraction method is proposed. The experimental framework and results are reported in Section 5. Some discussions are presented in Section 6. Finally, Section 7 provides our conclusions and plans for future work.

2. Related Work

A general taxonomy divides the baseline detection in three categories: curved text-line detection, scene text detection and handwriting text-line detection. Our aim is to improve handwriting line detection where the problem is not to detect text lines in a complex scene but rather to detect lines in scanned, historical, handwritten documents. Doing so involves some challenges such as complex layouts, irregular character sizes, varying skews, noise artifacts, and touching lines of text. As the present work is devoted to handwriting baseline detection, we shall try to focus on the state-of-the-art handwritten baseline detection. But, it must be said that in the literature on this topic, the terms detection and extraction are frequently used indistinctly. The most common taxonomy in literature is that presented by B. Gatos [16] where text baseline detection methods are classified into four categories:

a) Methods based on projection profiles. This technique [17, 18] assumes that the text lines are almost horizontal and parallel. To allow some text line deviation from the horizontal axis, the image is segmented into vertical strips and projections are performed for each one. Some morphological or blurring techniques are used in these methods to smooth the horizontal projections.

b) Methods based on the Hough transform. These methods work properly in printed text [19]. The Hough transform is a well-known tool used in document analysis that allows the finding of the line angle that crosses the most points in a set of points, starting from one point. These points of interest are usually the gravity centers of the connected components [20, 21] or the local minima points [22] of the connected components.

c) Clustering methods. The aim is to cluster the basic building elements of text such as pixels, connected components or other structures detected from the contours into sets that correspond to lines, including local minima. In [23], a graph has been built with the connected components as

vertices and using the distances between components as edges weights. In [24], an adaptive local connectivity map is used. A new image is obtained by finding the sum of the intensities of the neighbors pixels in the horizontal direction. A grouping method is then used to cluster the connected components. In [25], the maximally stable external region (MSER) algorithm is used rather than conventional binarization algorithms. The scale and local orientation of connected components (CCs) is used to infer a line spacing for each CC.

d) Dynamic programming based methods try to segment the lines by finding an optimal path which crosses the image from the left to the right edge [15, 14]. Saabni transforms the input image into an energy map and determines the seams that cross between the text lines. Nicolaou takes the assumption that for each text line, a path exists which crosses the image from left to right.

Another common taxonomy divides the methods into two categories: top-down and bottom-up [26]. Top-down analysis consists to split the whole page into a set of subunits and continues this splitting until having pieces such as text lines. On the other hand, bottom-up analysis starts by merging the smallest primitives (CCs, superpixels, etc.) and continues merging until obtaining the page components at the highest level. It is easy to find other methods in the literature, most of them based on heuristic techniques or on a combination of other techniques. A good survey can be found in [12], [27] and [28] along with a more detailed description of methods for text line detection and segmentation.

3. Baselines Detection Algorithm

The purpose of line detection algorithms is to mark where the text lines are in page images. The method proposed here is designed with the aim to resist the noise artifacts that usually appear in historical manuscripts.

The baseline detection problem can be formulated as a clustering problem over a set of interest points. Let P be a set of points, we are looking for a

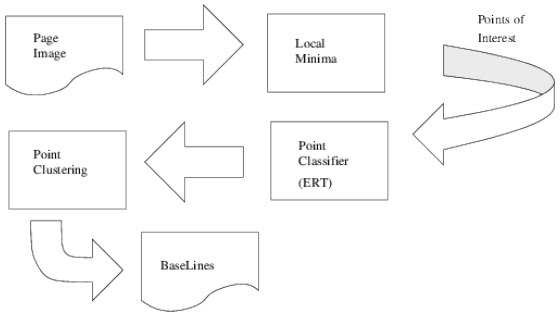

Figure 1: Work Flow. From the page image, the local minima points are obtained. In order to avoid points from noise artifacts, these points are classified as belonging, or not to a baseline. Then, points are clustered using a top-down clustering tree algorithm (see Fig. 6). Each resulting class contains minima points belonging to a baseline.

partition W so that $W = \{C_1, C_2, .., C_n\}$ where $C_1 \cup C_2 \cup ... \cup C_n = P$ and $C_i \cap C_j \in \emptyset$ $(\forall i, j \wedge i \neq j)$. The polyline formed by joining each couple of points in a cluster C_i, previously ordered along the x-axis, is considered a baseline. In this work, points that are candidates for belonging to baselines are the local minima of the image text edges (see Figure 2). It is normal to find among these points some belonging to descenders, noise spots, borders, etc. To deal with this, a classification system has been designed to detect those points which really belong to baselines. From these points, a clustering algorithm is used to cluster the points into lines. Figure 1 gives the workflow for this process. A modification of DBSCAN[29] is used in the present work. It must be said that this algorithm does not need to know the number of clusters in advance.

3.1. Interest points

Interest points are those points susceptible to belonging to baselines. Local minima points, obtained from the text edges are chosen as interest points (see Figure 2 for a contour example). After a blur and a binarization by using the well-known global Otsu algorithm, the contours of the image foreground are obtained. Then, using an analysis window centered on each point from the contours, the local minima are obtained. Since the local minima points are

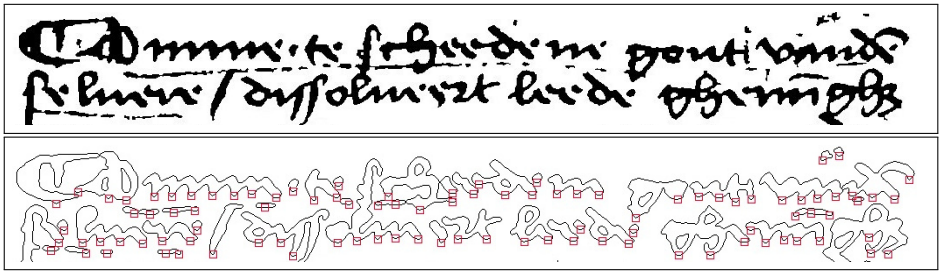

Figure 2: Top: A binarized image detail. Bottom: The text contour with local minima points.

usually obtained from noisy images, an automatic point classifier is needed to bring robustness in front of the noise and other kinds of artifacts found on these pages. Therefore, the first problem to solve is the local minima points classification. In this work, a forest of *Extremely Randomized Trees* (ERT)[30] has been used as a classifier. The ERT input consists of a downsampled image window context around the point to be classified. The window geometry is empirically set. Three randomly chosen pages are manually annotated and then are used to train several ERTs, one for each training page and window geometry. These ERTs are used to classify all of the minima points. Results for the ALCARAZ corpus can be seen in Figure 3. It must be observed that this parameter is not very sensitive if it is chosen in a meaningful range. When the context becomes insufficient, the error quickly starts to grow. Similar behaviors are observed in the whole corpora set. In the present work, a 100×50 pixels window context around the point of interest has been taken. Then this context image was downsampled into a 50×30 pixel image. .

3.1.1. Extremely Randomized Trees (ERT) Forest Classifier

This is a tree-based ensemble method for supervised classification introduced in 2005 by P. Geurts et al. [30]. The term came from the random decision forest that was firstly proposed by Tin Kam Ho [31] in 1995. This method combines the bagging idea with the random selection of features in order to get a low variance. This classifier was selected because it works quite well with a few labeled samples and is fast at estimating its parameters and at classification [32].

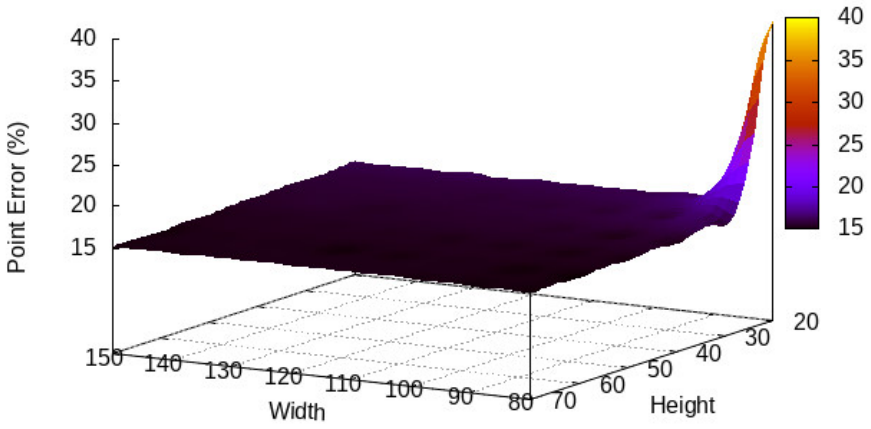

Figure 3: Point error classification for the ALCARAZ corpus. Each point represents the average classification error of three ERTs on a single, randomly chosen page that was trained for window geometry.

3.1.2. Ground truth

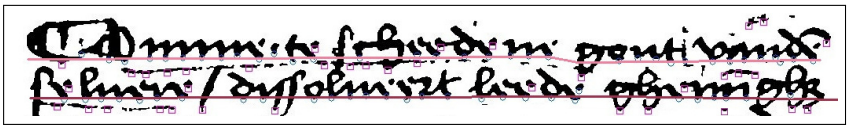

Figure 4: An example of the labeling process. From the manual baselines, points which are close enough to them are classified as belonging (blue circles) and those which are too far are away are classified as not belonging (violet squares).

Two types of ground truth are needed: one for training the ERT and another for baseline evaluation purposes. The first one is obtained automatically from the second. The labeled points used to train the ERTs were estimated from manual baselines. A simple method is used to classify the interest points as belonging to a baseline or not. The method consists of labeling those points which are close enough to a baseline of reference, those whose distance is less

than a predetermined threshold as belonging to a baseline, and those which are too far from that threshold so as to not belong (see Figure 4 for an example). As our baseline detection method requires relatively little training data (a single page) we developed a graphical tool to achieve a fine adjustment (see Figure 5). In this way, after choosing the upper and lower distance limits, we can easily move the baselines up or down slightly as well as any of its composing segments to automatically correct the point labeling.

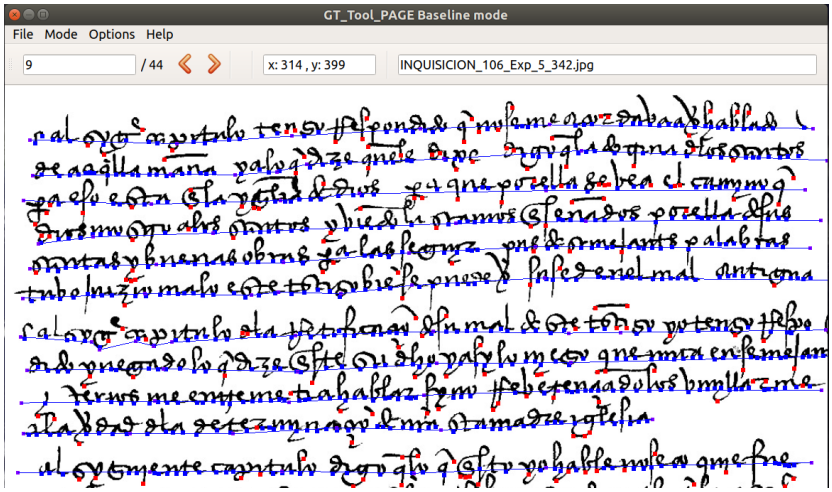

Figure 5: A graphical tool for fine point classification adjustment. The blue points are those classified as belonging to the baseline while the red points are classified as not belonging. Minima point coordinates cannot be changed but by moving a baseline (or a segment of it) the point labels change automatically.

The average of points per page used to fit the baseline to the text lines was 3.6 in the ICDAR13, 5.5 for ALCARAZ, and 4.3 on the HATTEM corpus. These corpora will be explained in detail in Section 5.1.

3.2. Clustering points into baselines

Excluding the interest points classified as not belonging to the baselines, a top-down clustering algorithm based on a modified DBSCAN (density-based spatial clustering of applications with noise) [29] is used to get baselines (see

Figure 6). DBSCAN exploits the notion of density reachability. A point, p_1, is reachable from another point, p_2, if its Euclidean distance is lower than a given distance, ε. A point belongs to a cluster if it has a sufficient number of reachable points in its neighborhood. This requires two parameters: the distance to be used as a boundary for the neighborhood, ε, and a minimum number of reachable points in a neighborhood in order for that to be considered a dense neighborhood, K. It must be highlighted that this algorithm does not need to know the number of clusters.

In order to take advantage of the a priori knowledge of the quasi-horizontal distribution of the local minima points belonging to a baseline, the metrics of the original DBSCAN algorithm were changed to use the Mahalanobis distance instead of the Euclidean one (see Equation 1).

$$d_m(\vec{p_1}, \vec{p_2}) = \sqrt{(\vec{p_1} - \vec{p_2})^T \Sigma^{-1} (\vec{p_1} - \vec{p_2})} \tag{1}$$

where:

$$\Sigma = \begin{Bmatrix} \varepsilon_x & 0 \\ 0 & \varepsilon_y \end{Bmatrix}$$

And $\varepsilon_x = \varepsilon$ and $\varepsilon_y = \lambda\varepsilon$

To get the baselines, an initial first partition is obtained by using the DBSCAN algorithm. Then a criteria function is used to decide if each class of cluster is a leaf or not, in which case it will need to be split. Every time a class is proposed to be split, the DBSCAN neighbors area managed by ε is reduced and the modified DBSCAN is applied to this cluster point set (see Figure 6). The criteria function to stop splitting clusters is based on the analysis of the slopes of the straight lines joining two consecutive points once those points are ordered along the x-axis. A real graphic example can be seen in Figure 7.

4. Line extraction

As line extraction is an open interesting problem, a *line extraction* algorithm was implemented. The purpose of line segmentation algorithms is to divide the

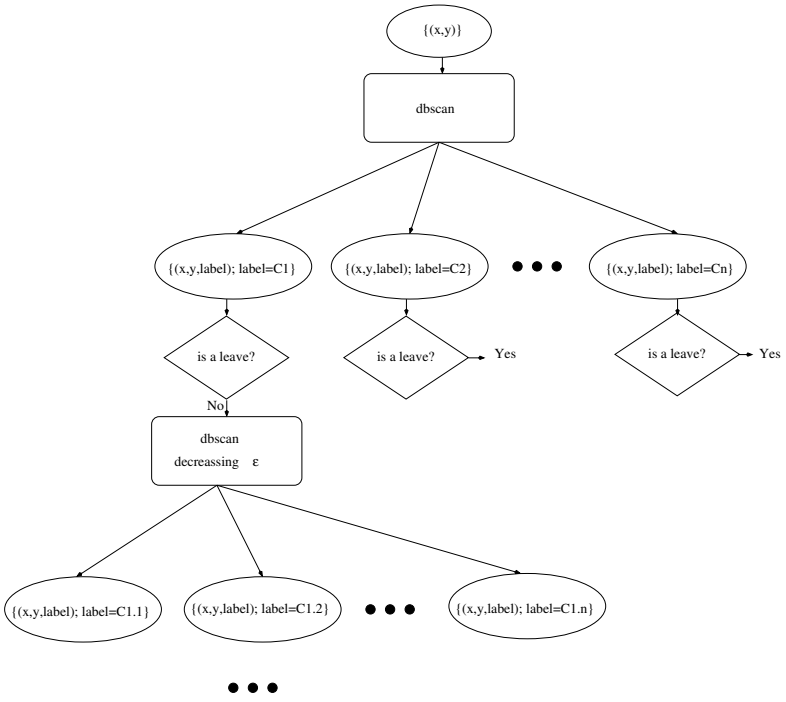

Figure 6: Scheme of the top-down clustering tree algorithm. Each set of interest points (tree nodes) is split using the modified DBSCAN. This process is repeated at each node if the node contains points belonging to a more than one single baseline.

image into areas in which each one contains a single text line.

Using baselines as restrictions, an algorithm was designed to find an optimal path from the left to the right side of the image between every pair of consecutive baselines (see the example in Figure 8). First, these areas of the image are de-
200 skewed [33]. A directed weighted graph is built from this image taking pixels as nodes. Five edges are inserted for each node (8-connected but taken out the back edges), except for the ones corresponding to the last image column. The edge weights are the sum of the pixel complement values for the involved nodes. That way, if both are white, the weight will be 0 while if both are black, the
205 weight will reach its maximum value. Edge weights are calculated as follows:

$$w(v(x,y), v(x',y')) = Inv(I_{x,y}) + Inv(I_{x',y'})$$

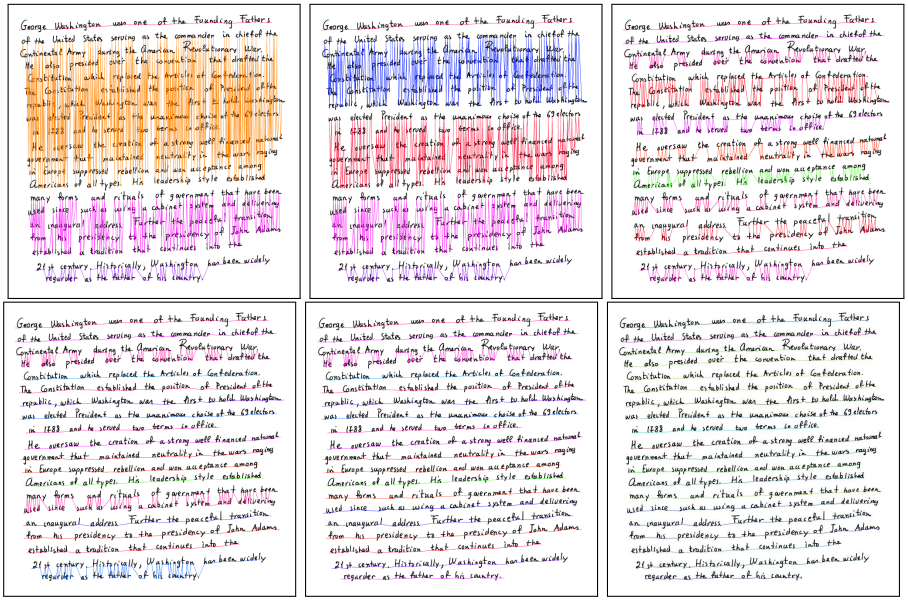

Figure 7: A graphic example of the top-down clustering algorithm. From top to bottom, and from left to right: In the first image, a clustering proposition after the application of the first modified DBSCAN. It must be noted that the first cluster only contains a baseline while the rest of the clusters need to be re-clustered by previously decreasing the neighborhood area, ε. After re-clustering, the second cluster (the yellow area) is split into two smaller clusters (as seen in the second image). These new clusters need to be re-clustered again because both contain points belonging to more than one baseline. This process is repeated until all contain points from a single baseline.

where $v(x, y)$ is a function that returns the vertex associated with the image pixel (x, y) and the function $Inv(I_{x,y})$ returns the complementary value of the pixel at the position (x, y). An initial vertex is chosen to be on the left side and in the middle of both baselines. Any vertex corresponding to a pixel in the last column is taken as a final vertex. The well-known Dijkstra algorithm is used to find the shortest path between the initial node to one of the finals. These paths are used as borders between the text lines areas. An example of a page with detected baselines and the result of segmenting this page by using them can be seen in Figure 8.

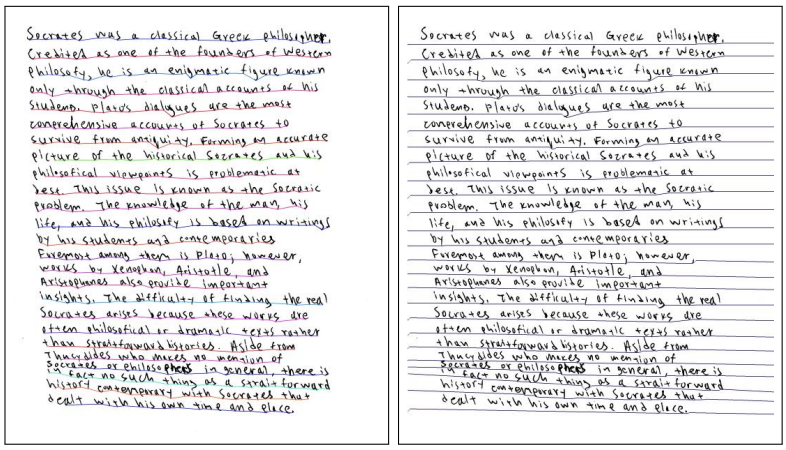

Figure 8: On the left: An example of detected baselines as presented for one page of the Icdar'13 contest corpus. On the right: The segmentation seen by using these baselines as restrictions.

5. Experimentation

5.1. Corpora

Alcaraz *corpus*

This corpus is a small set of documents from the Spanish Inquisition process against Pedro Ruiz de Alcaraz which took place from 1534 to 1539 [34]. It is composed of 44 pages with a total of 1,731 lines that were produced by a single writer. The whole manuscript was written in ancient Spanish and shows notable age degradation. An example of this corpus can be seen in Figure 9.

Hattem *corpus*

Hattem corpus is a manuscript from the 15th century, composed of 572 pages. Most of it is written in Dutch while some is in Latin and French. It is a prose translation of the Secretum Secretorum (which is a Latin translation of an Arabic encyclopedia on government, health, astrology, and alchemy) which was transcribed for the Pope [35]. Figure 10 shows some examples of pages from the corpus. The corpus set used here is the same as used in [13]. This corpus is composed of 40 pages (1,542 text lines) which are not consecutive pages from

Figure 9: Example manuscript pages from the ALCARAZ corpus.

the book. They were selected from the complete collection by an expert who was given the criteria of providing a reduced but representative set of all page formats that appear in the book.

Figure 10: Example manuscript pages from the HATTEM corpus.

In this work, manual baselines and a perfectly fitted polygon surrounding each text line were used (as kindly provided by the authors of [13]). To obtain the polygons, they used the segmentation technique presented in [20] and during a post-process, a manual correction was performed. These polygons were used

for the aim of comparison with the ICDAR'13 segmentation context.

ICDAR *2013 Handwriting Segmentation Contest corpus*

This corpus is the same as used at the ICDAR 2013 segmentation contest [36]. The corpus is composed of 350 image pages of manuscript text with their associated ground truth, written in three different languages: Latin, Greek, and Bangla. The corpus is presented split into two partitions: one for training purposes and the other for testing.

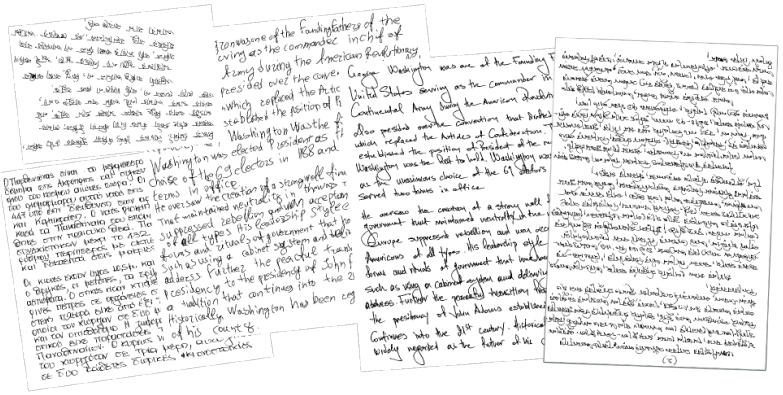

Figure 11: Example pages from the ICDAR'13 contest corpus.

The text images are presented as black and white handwritten document images produced by different writers. The ground truth were raw data image files with zeros for the background and a positive integer value, each one corresponding to the foreground of a segmentation region. Some examples of this corpus can be seen in Figure 11.

5.2. Experimentation and Results

5.2.1. Baseline detection measures

A metric proposed by Tobias Grüning [37] , based on the classic precision, recall, F-measure, is used here.

Let $\mathcal{G} = g_1, ...g_M$ be the baseline ground truth and let $\mathcal{R} = r_1, ..., r_K$ be a baseline hypothesis. As each baseline can contain an arbitrary number of points,

each polyline is "blown up" to take in all points along the baseline. Let $\mathcal{G}^*$ and $\mathcal{R}^*$ be the blow up sets for $\mathcal{G}$ and $\mathcal{R}$ respectively. A tolerance value t_i has to be calculated for each polyline in the ground truth g_i because the baselines are not unique. To estimate the tolerance values, the author of this algorithm followed the approach in [38] and used the distances of adjacent ground truth baselines. In his opinion, the tolerance values are not crucial if they are in a meaningful range because the metric is not highly sensitive to them.

To calculate the precision and recall, a counting function is needed. The counting is calculated from one ground truth line, $g_i^* \in \mathcal{G}^*$ and second $r_j^* \in \mathcal{R}^*$. Actually, what is being done is counting through the whole set of found lines, $\mathcal{G}^*$. In this way, the recall value only represents how much of the ground truth line is found no matter whether it is under- or over-segmented. The recall for ground truth lines is calculated as follows:

$$recall(g_i^*, R^*) = \frac{cnt(g_i^*, \mathcal{R}^*, t_i)}{|g_i^*|} \quad i = 1, ..., |\mathcal{G}^*| \tag{2}$$

On the other hand, precision must be a measure of how accurate the hypotheses are. For a hypothesis and a ground truth set $(\mathcal{G}^*, \mathcal{R}^*)$, the proposed precision measure is calculated by searching for the best matching pairs, $\mathcal{M}$, on the cross matrix $C \in \mathbb{R}^{M,K}$ where $c_{i,j}$ is calculated as follows:

$$prec(g_i^*, r_j^*) = \frac{cnt(r_j^*, g_i^*, t_i)}{|r_j^*|} \tag{3}$$

From the matching pairs, $\mathcal{M}$, the recall and precision at a page level are calculated as follows:

$$recall_{page}(\mathcal{G}^*, \mathcal{R}^*) = \frac{\sum_{g^* \in \mathcal{G}^*} recall(g^*, \mathcal{R}^*)}{|\mathcal{G}^*|} \tag{4}$$

$$prec_{page}(\mathcal{M}, \mathcal{R}^*) = \frac{\sum_{(g^*, r^*) \in \mathcal{M}} prec(g^*, r^*)}{|\mathcal{R}^*|} \tag{5}$$

5.2.2. Segmentation measures

Also in this paper, the same performance evaluation tool of ICDAR 2007 Handwriting Segmentation Contest [36] was used. Its metrics are based on the number of matches between the areas detected and the areas in the ground truth [39].

Let I be the foreground points inside the set of all images; G_j and R_j be the foreground points for the j text line ground truth and the hypothesis, respectively; and $T(s)$ be a function that counts the elements of set s. $MatchScore(i, j)$ represents the matching results of the i ground truth region and the j result region as follows:

$$MatchScore(i, j) = \frac{T(G_j \cap R_i \cap I)}{T((G_j \cup R_i) \cap I)}$$

A *one-to-one* match between a hypothesis region, i, and a ground truth region, j, pair is only considered if the matching score is equal to or above a specified acceptance threshold, T_a. In the present work, the T_a was set to 95%, as it was set in the ICDAR 2013 for line segmentation evaluation.

Let N be the count of ground-truth elements, M the count of hypothesis elements, and $o2o$ the number of *one-to-one* matches. The detection rate (D_R) or recall and precision (R_A) are calculated as follows:

$$D_R = \frac{o2o}{N}, \quad R_A = \frac{o2o}{M}$$

A performance metric, the harmonic mean of the precision (R_A) and recall (D_R), the traditional $FMeasure$ (FM) is calculated in this way:

$$FM = \frac{2\ D_R\ R_A}{D_R + R_A}$$

This metric seems to be robust and well established since it has been used in different fields and in similar contests including ICDAR 2007[40], ICDAR 2009 [41] and ICFHR 2010[42], and only depends on an acceptance threshold T_a.

The results have been calculated by using the same evaluation software provided by the ICDAR 2013 contest.

5.2.3. Experimental work

The number of trees in the ERTs was fixed at 100 and K was set experimentally at 3. The initial ε value was set automatically.

Ground truth

The ground truth for Alcaraz and Icdar'13 was manually annotated at the baseline level by the authors of this work. The Icdar'13 segmentation contest pixel-level ground truth is publicly available. For Hattem, the ground truth used (manual baselines and polygons) is the same as the one used by V. Romero et al. in [13] (kindly provided by the authors). Polylines per baseline were composed of 5.5, 4.3, and 3.6 points on average for Alcaraz, Hattem and Icdar'13 while the average lines per page was 39.3, 39.8, and 17.7, respectively. This is an indicator of the amount of work needed to label the manual baselines.

One page training experiments: Taking one in

For every corpus, an ERT was trained for each of its pages using the labeled points classified from manual baselines. A line detection experiment was carried out for each ERT, using the remaining pages only for testing. The results in Table 2 are presented as the average and standard deviation of all experiments.

		Original metrics			Mahalanobis metrics		
		Precision	Recall	FM	Precision	Recall	FM
Alcaraz	Average	0.70	0.81	0.75	0.98	0.97	0.97
	Std. dev.	0.027	0.019	0.023	0.003	0.002	0.002
Hattem	Average	0.71	0.43	0.54	0.99	0.85	0.92
	Std. dev.	0.006	0.057	0.047	0.002	0.019	0.011
Icdar	Average	0.80	0.80	0.80	0.99	0.96	0.97
	Std. dev.	0.184	0.175	0.183	0.02	0.07	0.05

Table 2: Detection results. Average results for 44 (Alcaraz), 40 (Hattem), and 150 (Icdar13) single-page trained systems. All systems used Dbscan original metrics and Mahalanobis metrics.

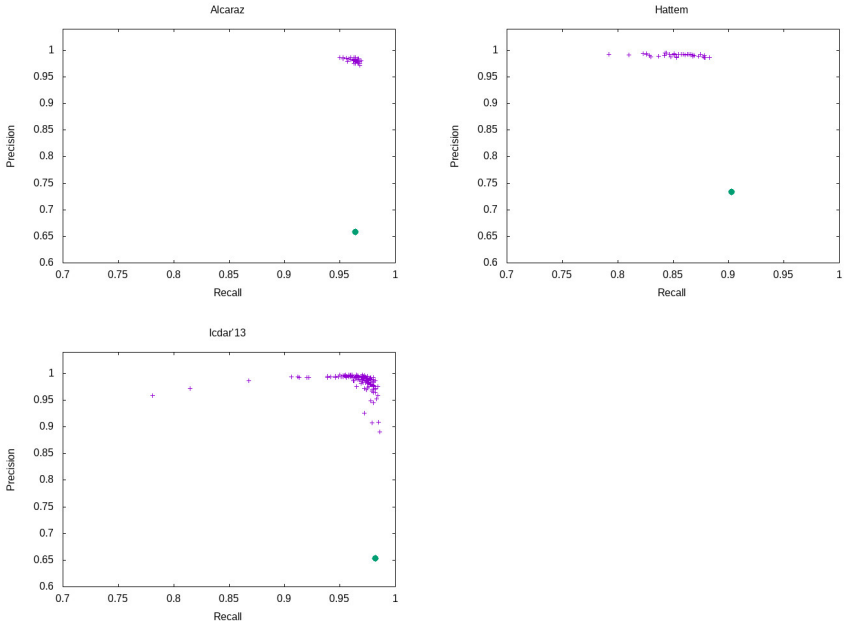

Figure 12: ALCARAZ, HATTEM and ICDAR13 skater plots. Each point represents the precision recall for an ERT single-page trained system. The large green dot is the result for a zero-page trained system.

The low standard deviations values must be noted. This is a clue as to the small amount of training data needed to train and the stability of the system. In the skater plots (see Figure 12), there is a point for each single-page trained system. The biggest dot (in green) represents a system that was trained with zero pages. All candidate points were taken as belonging to baselines, including those that were obtained from descenders, noise, or any kind of artifact. The results for zero-page trained systems presents a high recall value due to the fact that the whole set of points belonging to baselines are included, covering most of them but also introducing false positives and over-segmentation. It must be said that a classifier which brings low performance can erroneously classify points as noise, thus harming the clustering process.

Increasing training experiments

To check the influence of the amount of training data for ERTs, some incremental training experiments were performed. This was first done without any classifier, considering all points as belonging to baselines (zero-page training). Then an ERT was trained with the first page, then the first two pages, the first three pages, and so on. In these experiments, the corpora were partitioned into training sets and test sets (see Table 3 for details).

Corpus	train	test
Alcaraz	30	14
Hattem	30	10
Icdar'13 *	40	110

Table 3: The number of pages for training sets and for test sets. *Due to the very different nature of the pages, a shuffle was applied before segmenting this corpus into sets.

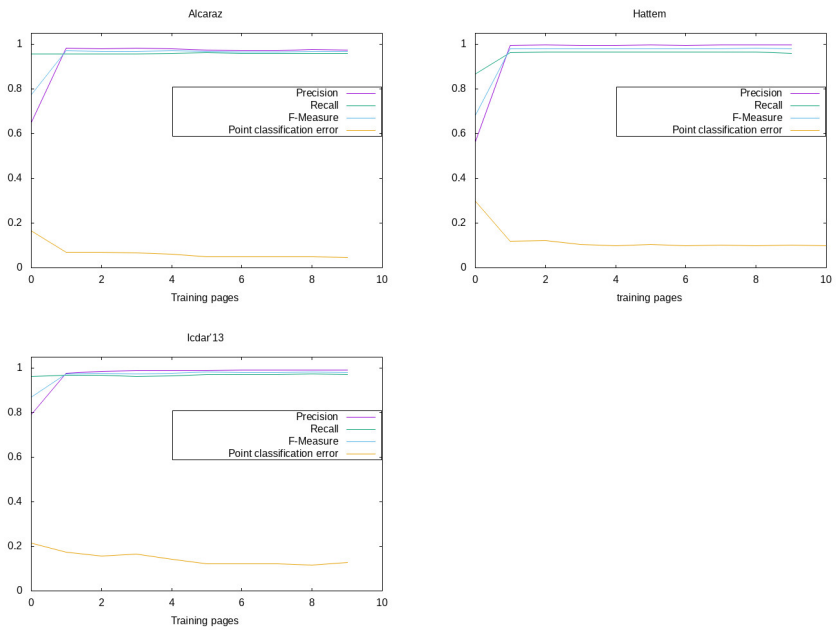

Figure 13: Baseline precision, recall, and F-measure results for ALCARAZ, HATTEM, and ICDAR13 systems trained with an increasing number of pages.

The observed behaviors (see Figure 13) are similar along all three corpora. All of the tests performed presented a prominent increase in precision from the zero-page trained systems to the trained ones. The biggest increment is achieved in going from zero pages to those with a single page. From there, the results remain relatively stable. The relative increments produced by training with a single page with respect to the zero-page trained ones is 25.6% on F-Measure, 51.5% on precision, and 2.1% on recall for ALCARAZ; a 43.6%, 77.1%, and 11.2% for HATTEM; and 12.0%, 23.5%, and 0.6% for ICDAR'13. Similar behavior was found in the interest points classification error results.

Extraction experiments

As the ground truth is available at pixel level for HATTEM and ICDAR'13, kindly provided by authors of [13] and by the organizers of the ICDAR'13 segmentation contest, respectively, some line segmentation experiments were carried out for the sake of comparison. Results of these comparisons can be seen in Table 4.

The first row shows the best result obtained by Romero et al. They used cross-validation, dividing the corpus into eight blocks of five pages. The next row shows the optimistic values obtained by the method proposed in Section 4 using the manual baselines provided by Romero et al. Finally, the average and standard deviation for the 40 single-page trained ERT systems are included. The difference between using manual baselines and those provided automatically by the system (with one training page) has an F-measure of 6.7 points (7.9% relative).

As a polygon including a text line is not unique and the ground truth is labeled at the pixel level, this forces the measure evaluator to include some kind of tolerance. HATTEM images are filled with noise, decorations, dropped capitals, etc. Therefore, a polygon including a text line could be considered to be unmatching because it includes enough noise to overpass the tolerance value. The authors of the present work deleted all dropped capitals from the HATTEM images and the results of this can be seen in Rows 4 and 5 of Table 4.

		DR(%)	RA(%)	FM(%)	o2o
HATTEM	V.Romero et al. [13]	82.0	82.0	82.0	1306
	Manual baselines	83.47	83.95	83.71	1329
	Average	75.66	78.55	77.07	1202.8
	Standard deviation	1.38	1.14	1.24	25.26
	Average no DropCap	83.47	86.62	85.05	1327
	Std. dev. no DropCap	1.6	1.4	1.4	28.9
ICDAR'13	Contest winner	98.68	98.64	98.66	2614
	Average	98.72	98.60	98.66	2615.3
	Standard deviation	0.40	0.36	0.32	10.6

Table 4: HATTEM and ICDAR'13 contest corpus segmentation results. Row 1: Results obtained by Romero et al. Row 2: The results of using the manual baselines provided by Romero et al. as restrictions on the line extraction algorithm. Rows 3 and 4: The average and standard deviation for the extraction algorithm using the baselines provided by 40 single-page trained ERT systems. Rows 5 and 6: The same results after manually taking out the dropped capitals. Rows 7, 8, and 9: For the ICDAR'13 contest, the winning results as well as the average and standard deviation for the results based on the baselines provided by 150 ERT systems.

For the well-known, publicly available ICDAR'13 segmentation contest for which ground truth is available at the pixel level, some line extraction experiments were carried out. The contest winner result is presented in Row 7 of Table 4. It must be noted that the corpus for this contest was quite irregular with significant variations in handwriting, alphabet, language, etc. while the results remained quite stable.

6. Discussion

The well-known global Otsu algorithm works pretty well even with degraded, aged documents due to this algorithm bringing with it a binarization that usually does not lose the foreground while emphasizing some noise artifacts. The need for binarization is not for document restoration or enhancement but rather to find the borders between the foreground and background. The price paid for this

is the enhancement of the noise artifacts. The majority of the points obtained from the noise artifact are rejected in the next phase, point classification.

For the case of page images that suffer from a lack of contrast, a preprocess must be carried out to avoid losing parts of the line text. Although, the most disturbing noise artifact is the bleed-through where the system can label them as belonging to baselines (see Figure 14 A).

The proposed method is quite adaptive to deviations over the horizontal as skew and slope. Each baseline segment is composed of two points and the angle allowed for these segments is up to 65^o. If a cluster contains segments over this value, the cluster is determined as having more than one baseline and is then split. This value has been chosen empirically and is not very sensitive if chosen in a meaningful range. This will allow it to follow every text line if none of its segments exceeds this 65^o.

In Figure 14, some page examples with their automatically-detected baselines can be seen. On the top left, (A), is one image with no text but with bleed-through. This is one of the most disturbing artifacts for our method because it has no opportunity to detect if these lines belong to the foreground or if they are bleed-through. On the top right, (B), is a labeled page which presents some skew. On the bottom left, (C), is an image with two columns which have been labeled. The image did not have any previous layout information and had been treated as a single page. The image on the bottom right, (D), is an artificially undulated image where the baselines can be seen following this undulation.

The presented method complexity is $O(|pixels| + k\,n\,log(n))$ where n is the number of local minima baseline candidate points after ERT classification ($n << |pixels|$) and k is the number of clusters that must be re-split during the process, in which case $|n_c| < |n_p|$ where n_c is the candidate points number on the cluster c and n_p is the number of points in the cluster parent which the cluster came from (see Section3.2 for more details).

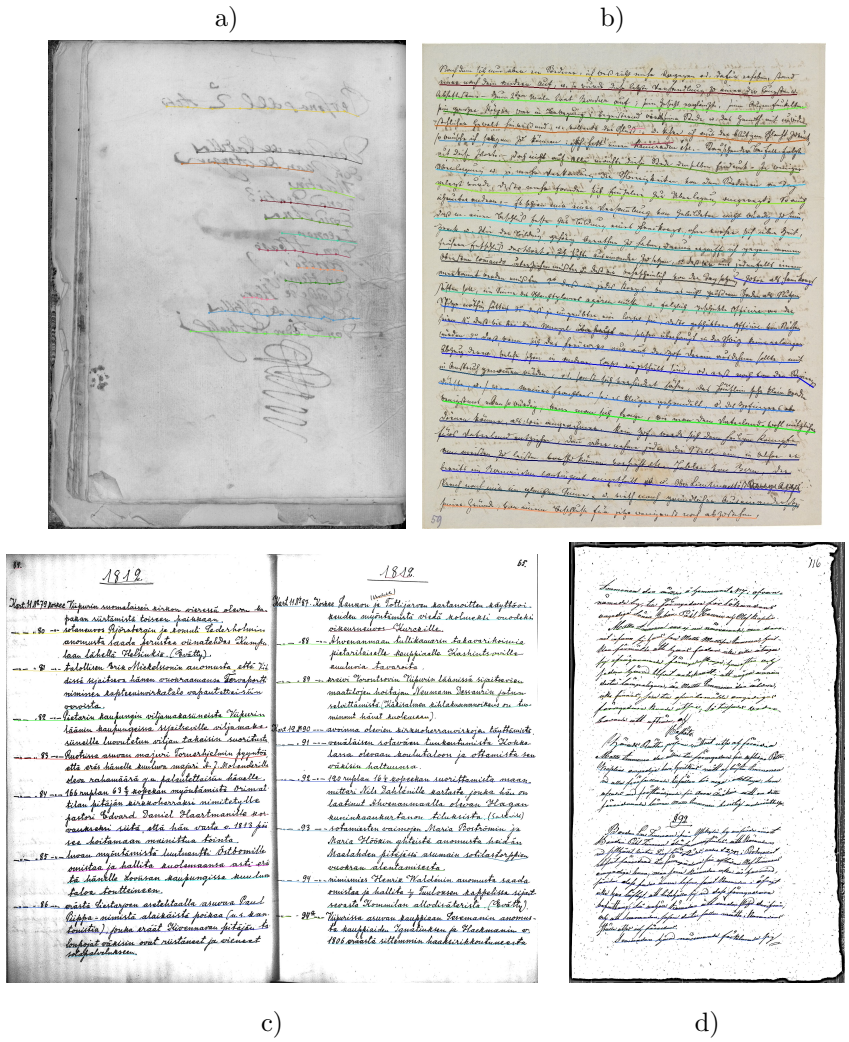

Figure 14: From top-left to bottom-right. A: Bleed-through detected baselines. B: Mistakes due to bleed-through. C: A two-column image. D: An artificially undulated page.

On the other hand, training the ERT classifier has a complexity of $O(ntree * mtry * nlog(n))$ where $ntree$ is the number of trees, $ntry$ the number of variables wanted to sample at each node, and n is the number of samples. The classification process is $O(nTree * log(n))$.

7. Conclusions

A fast baseline detection method has been presented here. The local minima of the text contours are considered to be interest points. An Extremely Randomized Trees forest is used to discriminate between points belonging to a baseline to those which do not. That makes the system robust to usual problems in historical documents like slant, slope, skew, noise or humidity spots. A modified version of Dbscan is used to cluster these points into baselines. The Mahalanobis distance is used as metrics to take advantage of the quasi-horizontal nature of the local minima points of the text lines. That bring a significative improvement.

A fast baseline detection method has been presented here.

The method seems to be resistant to the usual problems found in historical documents like slant, slope, skew, noise, and humidity spots. The local minima of the text contours are considered to be interest points. An extremely randomized trees forest is used to discriminate between the points belonging to a baseline between those which do not. A modified version of Dbscan is used to cluster these points into baselines. The Mahalanobis distance is used as a metric to take advantage of the quasi-horizontal nature of the local minima points of the text lines. That bring a significative improvement. The implementation in this work uses the page layout, if available, or the whole page if not. In the case of multiple text columns and no layout, a meaningful value for ε must be chosen (see Figure 14 C). The method presented shows how stable is, independently of the page chosen to train, and do not need especial hardware to run. It takes roughly five minutes to annotate the baselines of a page, three minutes to train an ERT, and three seconds on average to automatically estimate the baselines of each page on an Intel(R) Core(TM) i5-6400 CPU @ 2.70GHz computer.

Some extraction experiments were carried out for the aim of comparison. The use of baselines as a restriction had proved to be useful.

As a future work, we plan to use other features than image itself to classify the interest points, as for example geometric moment invariants in order to

reduce the point classification time. We are planing to use CNN's to enhance the images to avoid the lack of contrast that some page images present, in which cases, some of the text lines can be lost.

The software used in the present work can be freely download from `https://github.com/moisesPastor/baseLinePage`.

Acknowledgements

This work was partially supported by the project Carabela (PR[17]_HUM_D4_0059), sponsored by the programme "Ayudas a Equipos de Investigacion en Humanidades Digitales" of the BBVA Foundacion.

References

[1] M. Diem, F. Kleber, S. Fiel, T. Grüning, B. Gatos, ScriptNet: ICDAR 2017 Competition on Baseline Detection in Archival Documents (cBAD), This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 674943 (Jan. 2017). doi:10.5281/zenodo.257972.

[2] A. H. Toselli, V. Romero, M. Pastor, E. Vidal, Multimodal interactive transcription of text images, Pattern Recognition 43 (5) (2010) 1814–1825.

[3] D. Martín-Albo, V. Romero, A. H. Toselli, E. Vidal, Multimodal computer-assisted transcription of text images at character-level interaction, International Journal of Pattern Recognition and Artificial Intelligence 26 (5).

[4] V. Alabau, C. D. Martínez-Hinarejos, V. Romero, A. L. Lagarda, An iterative multimodal framework for the transcription of handwritten historical documents, Pattern Recognition Letters 35 (2014) 195–203, frontiers in Handwriting Processing.

[5] A. Fischer, A. Keller, V. Frinken, H. Bunke, Lexicon-free hand-written word spotting using character hmms, Pattern Recognition Letters 33 (7) (2012) 934–942.

[6] J. Puigcerver, A. Toselli, E. Vidal, ICDAR 2015 competition on keyword spotting for handwritten text documents, in: 13th International Conference on Document Analysis and Recognition (ICDAR), 2015, pp. 1176–1180.

[7] J. Puigcerver, A. Toselli, E. Vidal, A new smoothing method for lexicon-based handwritten text keyword spotting, 7th Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA) 9117.

[8] J.-M. S.Espana-Boquera, M.J.Castro-Bleda, F.Zamora-Martínez., Improving offline handwritten text recognition with hybrid hmm/ann models, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (4) (2010) 767–779. doi:10.1109/TPAMI.2010.141.

[9] O. Morillot, L. Likforman-Sulem, E. Grosicki, New baseline correction algorithm for text-line recognition with bidirectional recurrent neural networks, Journal of Electronic Imaging 22 (2) (2013) 023,028. doi:10.1117/1.JEI.22.2.023028.

[10] G. Meng, Z. Huang, Y. Song, S. Xiang, C. Pan, Extraction of virtual baselines from distorted document images using curvilinear projection, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 3925–3933. doi:10.1109/ICCV.2015.447.

[11] P. Yang, A. Antonacopoulos, C. Clausner, S. Pletschacher, J. Qi, Effective geometric restoration of distorted historical document for large-scale digitisation, IET Image Processing 11 (10) (2017) 841–853. doi:10.1049/iet-ipr.2016.0973.

[12] L.Likformant, A.Zahour, B.Taconet, Text line segmentation of historical documents: a survey., IJDAR 9 (2007) 123–138.

[13] V. Romero, J. A. Sánchez, V. Bosch, K. Depuydt, J. de Does, Influence of text line segmentation in handwritten text recognition, in: 2015 13th International Conference on Document Analysis and Recognition (ICDAR), 2015, pp. 536–540. doi:10.1109/ICDAR.2015.7333819.

[14] R. Saabni, A. Asi, J. El-Sana, Text line extraction for historical document images, Pattern Recognition Letters 35 (2014) 23–33. doi:10.1016/j.patrec.2013.07.007.

[15] B. G. A. Nicolaou, Handwritten text line segmentation by shredding text into its lines, in: Proceedings of the International Conference on Document Analysis and Recognition, ICDAR, 2009, p. 626630.

[16] B. Gatos, G. Louloudis, N. Stamatopoulos, Segmentation of historical handwritten documents into text zones and text lines, in: 2014 14th International Conference on Frontiers in Handwriting Recognition, 2014, pp. 464–469.

[17] S. S. Manivannan Arivazhagan, Harish Srinivasan, A statistical approach to line segmentation in handwritten documents, in: Proc.SPIE- The International Society for Optical Engineering, Vol. 6500, 2007, pp. 6500–6511. doi:10.1117/12.704538.

[18] V. Papavassiliou, V. Katsouros, G. Carayannis, A morphological approach for text-line segmentation in handwritten documents, in: 2010 12th International Conference on Frontiers in Handwriting Recognition, 2010, pp. 19–24. doi:10.1109/ICFHR.2010.11.

[19] Y.Y.Tang, C. S. C., C. Yan, M. Cheriest, Document analysis and understanding: a briev survey, in: ICDAR, 1991, pp. 17–31.

[20] G. Louloudis, B. Gatos, I. Pratikakis, C. Halatsis, Text line and word segmentation of handwritten documents, Pattern Recognition 42 (12) (2009) 3169 – 3183, new Frontiers in Handwriting Recognition. doi:https://doi.org/10.1016/j.patcog.2008.12.016.

[21] G. Louloudis, B. Gatos, I. Pratikakis, C. Halatsis, Text line detection in handwritten documents, Pattern Recognition 41 (12) (2008) 3758 – 3772. doi:https://doi.org/10.1016/j.patcog.2008.05.011.

URL http://www.sciencedirect.com/science/article/pii/S0031320308001775

[22] Y.Pu, Z.Shi, A natural learning algorithm based on hough transform for text lines extraction in handwritten documents, in: Proc. 6th Intl Workshop on Frontiers in Handwriting Recognition (IWFHR98), 1998, pp. 637–646.

[23] S. Nicolas, T. Paquet, L. Heutte, Text line segmentation in handwritten document using a production system, in: 9th Intl Workshop on Frontiers in Handwriting Recognition (IWFHR04), 2004, pp. 245–250.

[24] Z.Shi, S.Setlur, V. Govindaraju, Text extraction from gray scale historical document images using adaptive local connectivity map, in: Proc. 8th Intl Conf. on Document Analysis and Recognition (ICDAR05), 2005, pp. 794–798.

[25] H. I. Koo, Text-line detection in camera-captured document images using the state estimation of connected components, IEEE Transactions on Image Processing 25 (11) (2016) 5358–5368. doi:10.1109/TIP.2016.2607418.

[26] Y. Li, Y. Zheng, D. Doermann, S. Jaeger, Script-independent text line segmentation in freestyle handwritten documents, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (8) (2008) 1313–1329. doi:10.1109/TPAMI.2007.70792.

[27] S. Eskenazi, P. Gomez-Krämer, J.-M. Ogier, A comprehensive survey of mostly textual document segmentation algorithms since 2008, Pattern Recognition 64 (2017) 1–14. doi:10.1016/j.patcog.2016.10.023.
URL https://hal.archives-ouvertes.fr/hal-01388088

[28] S. Mao, T. Kanungo, "empirical performance evaluation methology and its application to page segmentation algorithms", IEEE trans. on PAMI 23 (3) (2001) 242–256.

[29] M. Ester, H. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of Second International Conference on Knowledge Discovery and Data Mining, 1996, p. 226231.

[30] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, Machine Learning 63 (1) (2006) 3–42. doi:10.1007/s10994-006-6226-1.

[31] T. K. Ho, Random decision forests, in: Proceedings of the 3rd International Conference on Document Analysis and Recognition, 1995, p. 278282.

[32] P. Geurts, G. Louppe, Learning to rank with extremely randomized trees, in: JMLR: Yahoo! Labs Learning to Rank challenge Workshop in the context of the 23rd International Conference of Machine Learning (ICML2010), 2011, pp. 49–61.

[33] M. Pastor, A. Toselli, E. Vidal, Projection profile based algorithm for slant removal, in: International Conference on Image Analysis and Recognition (ICIAR'04), Lecture Notes in Computer Science, Springer-Verlag, Porto, Portugal, 2004, pp. 183–190.

[34] A. H. N. (Spain), Inquisición, Toledo: proceso contra Pedro Ruiz de Alcaraz, por alumbrado, años 1534-1539, no. v. 1 in Inquisición, Toledo: proceso contra Pedro Ruiz de Alcaraz, por alumbrado, años 1534-1539, Archivo Histórico Nacional, 1534.

[35] J. A. Sánchez, V. Bosch, V. Romero, K. Depuydt, J. de Does, Handwritten text recognition for historical documents in the transcriptorium project, in: Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage, DATeCH '14, ACM, New York, NY, USA, 2014, pp. 111–117. doi:10.1145/2595188.2595193.

[36] N. Stamatpoulos, B. Gatos, G. Louloudis, U. Pal, A. Alaei, Icdar2013 handwriting segmentation contest, in: 12th International Conference on Document Analysis and Recognition., 2013, pp. 1434–1406.

[37] T. Grüning, R. Labahn, M. Diem, F. Kleber, S. Fiel, READ-BAD: A new dataset and evaluation scheme for baseline detection in archival documents, CoRR abs/1705.03311. arXiv:1705.03311.

[38] M. Murdock, S. Reid, B. Hamilton, J. Reese, Icdar 2015 competition on text line detection in historical documents, in: 2015 13th International Conference on Document Analysis and Recognition (ICDAR), 2015, pp. 1171–1175. doi:10.1109/ICDAR.2015.7333945.

[39] I. Phillips, A. Chhabra, Empirical performance evaluation of graphics recognition systems, IEEE Trans. of Patt. Analysis and Machine Intell. 21 (9) (1999) 849–870.

[40] B. Gatos, A. Antonacopoulos, N. Stamatopoulos, icdar 2007 handwriting segmentation contest, in: 9th International Conference on Document Analysis and Recogniition, 288, 2007, pp. 1284–12.

[41] B. Gatos, N. Stamatopoulos, G. Louloudis, icdar 2009 handwriting segmentation contest, in: 10th International Conference on Document Analysis and Recognitition, 2009, pp. 1393–1397.

[42] B. Gatos, N. Stamatopoulos, G. Louloudis, Icfhr 2010 handwriting segmentation contest, in: 12th International Conference on Frontiers in Handwriting Recognition, 2010, pp. 737–742.