Handwritten Text Line Segmentation by Shredding Text into its Lines

A. Nicolaou[1,2] and B. Gatos[2]

[1]*Department of Informatics,*
Technological Educational Institute of Athens,
Agiou Spiridonos Aigaleo 12210 Athens,
Greece
anguelos.nicolaou@gmail.com

[2]*Computational Intelligence Laboratory,*
Institute of Informatics and
Telecommunications,
National Research Center "Demokritos",
153 10 Athens, Greece
bgat@iit.demokritos.gr

Abstract

In this paper, we propose a novel technique to segment handwritten document images into text lines by shredding their surface with local minima tracers. Our approach is based on the topological assumption that for each text line, there exists a path from one side of the image to the other that traverses only one text line. We first blur the image and then use tracers to follow the white-most and black-most paths from left to right as well as from right to left in order to shred the image into text line areas. We experimentally tested the proposed methodology and got promising results comparable to state of the art text line segmentation techniques.

1. Introduction

Handwritten text line segmentation is still considered to be a major challenge in document image analysis. In a simple document analysis processing pipeline, it would follow image binarization and page segmentation, and precede word and character segmentation, character recognition etc. Since it is in the beginning of a pipeline of processing, it is very important to minimize errors so that next stages of pipeline get accurate input.

When dealing with handwritten text, line segmentation has to solve some obstacles that are uncommon in modern printed text. Among the most predominant are: skewed lines, curvilinear lines, fluctuating lines, touching and overlapping components (e.g. Fig. 1b,c), usually words or letters, between lines and irregularity in geometrical properties of the line, such as line width, height, leftmost most position, distance in between words and lines; such irregularity can be seen Fig. 1a.

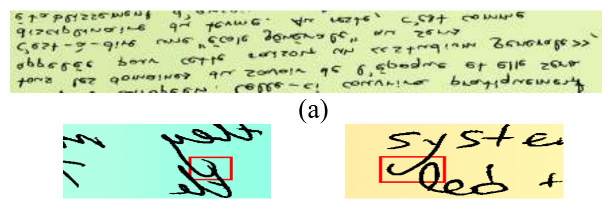


Figure 1. Line segmentation challenges.

There exist several methods for text line segmentation which are roughly categorized as follows. Smearing methods [1]: short white runs are filled with black pixels intending to form large bodies of black pixels, which will be considered as text line areas. Smearing methods can't deal well with touching and overlapping components. Horizontal projections [2]: a vector containing the sums of each image line is created. The local minima of that vector are assumed to be the projection of white areas in between lines, and the image is segmented accordingly. Horizontal projections can't deal well with skewed, curved and fluctuating lines. Hough transform [3] considers any image to compose of straight lines. It creates an angle, offset plane in which the local maxima are assumed to correlate with text lines. Hough transform has trouble detecting curved text lines. Bottom-up approaches: connected components or even pixels are connected to their close ones based on geometrical criteria to form text lines [2]. Other methods have also been proposed such as: repulsive attractive networks, stochastic methods and text line structure enhancing [2][4]. Due to many challenges in text line segmentation, although many methods have been proposed, the problem still remains open.

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.243

IEEE computer society

2. Proposed methodology

In this paper, we propose a new strategy for handwritten text line segmentation. We are motivated by the idea that a text image can be shred into strips along the white gaps in between text lines. Our approach is based on the topological assumption that for each text line, there exists a path from one side of the image to the other that crosses only one text line; this assumption applies to all images containing text in one column layout. A corollary of the above assumption is that for any pair of consecutive text lines, there exists a path from left to right that separates those two text lines. Our method tries to detect such paths and use them to shred the image into strips that contain one text line each. In many cases there is no line-separating path that traverses only white space between two consecutive lines (e.g. when components touch). To deal with such issues, and make our approach noise tolerant we blur the image before shredding it into text line strips. The image is shredded along the trajectories of many individual local minima tracers that traverse the blurred image from left to right and right to left. Once the images surface is separated into text line strips, we assign all connected components of the initial image to the appropriate text line. The proposed methodology consists of three main stages namely the preprocessing stage, the image shredding stage and the component assignment stage as can be seen in Fig. 2.

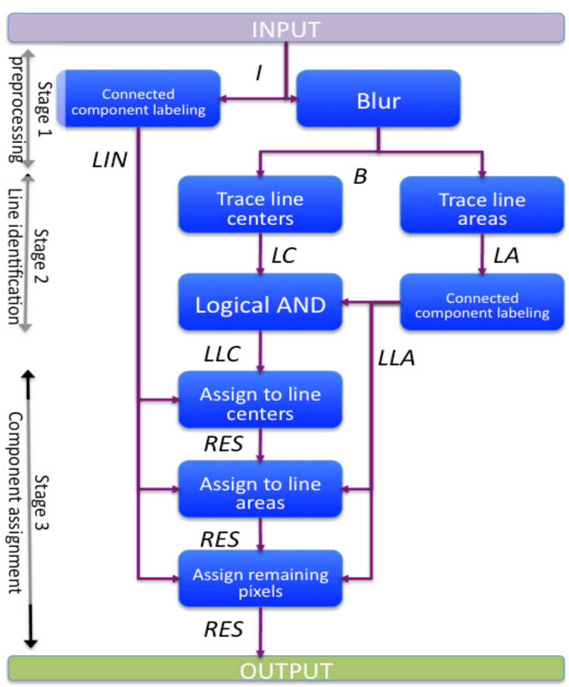


Figure 2. Proposed method flowchart

2.1 Stage 1: preprocessing

We assume the binary image *I(x,y)* which contains single column text information:

$$I(x,y) = \begin{cases} 1 & \text{if foreground pixel} \\ 0 & \text{if background pixel} \end{cases} \quad (1)$$

where $x \in [1, I_{width}]$ and $y \in [1, I_{height}]$. First we proceed to connected component labeling of *I(x,y)* and store the result in *LIN(x,y)*:

$$LIN(x,y) = \begin{cases} 0 & \text{if } I(x,y) = 0 \\ label & \text{if } I(x,y) \neq 0 \end{cases} \quad (2)$$

where $x \in [1, I_{width}]$, $y \in [1, I_{height}]$, $label \in \{1,2,3,\ldots,N\}$ and N is the number of connected components on the *I(x,y)*. Then we calculate the estimated letter height *LH* based on the histogram of the connected components height. We then blur *I(x,y)* as follows:

$$B(x,y) = \sum_{i=-BW/2}^{i=BW/2} \sum_{k=-BH/2}^{k=BH/2} I(x+i, y+k) \quad (3)$$

where *B(x,y)* is the blurred image, *BW=LH*8* is the bluring window width and *BH=LH*0.8* is the bluring window height. We set *BW* with the intent to blur-out white spaces in between consecutive words in a line and *BH* with the intention to blur-out letters in a line preserving white gaps between two consecutive lines[5]. In Fig. 3 we can see *B(x,y)*.

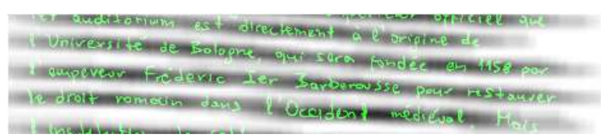


Figure 3. Part of an indicative B(x,y) with I(x,y) superimposed as green.

2.2 Stage 2: Image shredding

In this stage we shred the images surface into strips corresponding to text lines.

2.2.1 Tracing line areas: For each pixel in the left-most column of *B(x,y),* there exists a path along the whitest pixels that leads to the right-most column of *B(x,y)* without crossing any text lines. To calculate such paths, we recursively define the functions $Tr_{k,B}(n)$ which will be referred to as tracers:

$$Tr_{k,B}(1) = k$$

$$Tr_{k,B}(n+1) = \begin{cases} Tr_{k,B}(n) - 1 \text{ if } B(n, Tr_{k,B}(n) + BH/2) > B(n, Tr_{k,B}(n) - BH/2) \\ Tr_{k,B}(n) \quad \text{ if } B(n, Tr_{k,B}(n) + BH/2) = B(n, Tr_{k,B}(n) - BH/2) \\ Tr_{k,B}(n) + 1 \text{ if } B(n, Tr_{k,B}(n) + BH/2) < B(n, Tr_{k,B}(n) - BH/2) \end{cases} \quad (4)$$

We store all possible tracers on *B(x,y)* in a binary image *LA(x,y)* as 0s.

$$LA(x,y) = \begin{cases} 0 \text{ if } \exists \text{ k} : \text{Tr}_{\text{k,B}}(x) = y \\ 1 \text{ in all other cases} \end{cases} \quad (5)$$

LA(x,y) will contain the strips of text line areas as 1s and their separation points as 0s. We then apply the same process tracing white paths from right to left and draw the trajectories on *LA(x,y)* as well, in order to reduce the occurrence of missed lines. Since each tracer depends only on *B(x,y)* and its previous value, once two co-directional tracers pass from the same point, their trajectories will be identical as can be observed in Fig. 4. As it can be easily concluded from (4), the greatest ascend or descend angle of a tracer is 45^{o} and it is expected that all tracers will reach the end of the image in the same number of steps.

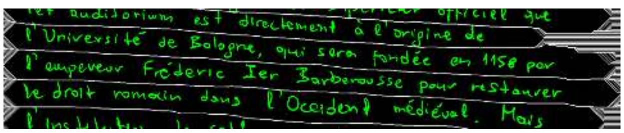


Figure 4. Part of an indicative *LA(x,y)* with *I(x,y)* superimposed as green.

2.2.2 Labeling line areas: Once we have drawn white path tracers on *LA(x,y)* we calculate 4neighbor connected components and store them *LLA(x,y)*

$$LLA(x,y) = \begin{cases} 0 \text{ if } LA(x,y) = 0 \\ label \text{ if } LA(x,y) \neq 0 \end{cases} \quad (6)$$

where $x \in [1, I_{width}]$, $y \in [1, I_{height}]$, $label \in \{1,2,3,...,N\}$ and N is the number of connected components in *I(x,y)*. As we can see in Fig. 5, there are components representing text line areas (colored areas in figure) and very small components (the black areas in figure) which created as the tracers converged to the local minima. Since all components in *LLA(x,y)* are supposed to represent different text lines, we filter out the components that have less pixels than LH^2 (there can be no line smaller than a letter).

2.2.3 Tracing line centers: In this step we run tracers on the inverted blurred image *–B(x,y)* and draw their trajectories on *LC(x,y)* as follows.

$$LC(x,y) = \begin{cases} 1 \text{ if } \exists \text{ k} : \text{Tr}_{\text{k,-B}}(x) = y \\ 0 \text{ in all other cases} \end{cases} \quad (7)$$

Figure 5. Part of an indicative *LLA(x,y)* with *I(x,y)* superimposed as bright green. White pixels have the null label, black pixels were filtered out and colored pixels represent different line areas.

LC(x,y) contains the black-most paths from left to right, this information will allow a more refined treatment of overlapping components and will help us identify very accurately touching components. Tracers in *LC(x,y)* also tend to correlate with the center of the strip between the base and the median lines in most documents. Since *LC(x,y)* will be used for optimization and is not as crucial as in *LA(x,y)*, we omit right to left tracing and trace only from left to right. In Fig. 6 we can see that all letters intersect with one and only one line.

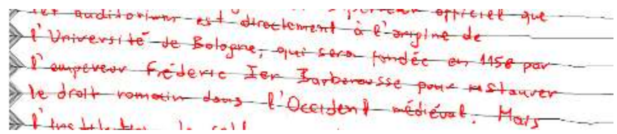


Figure 6. Part of an indicative *LC(x,y)* with *I(x,y)* superimposed as red.

2.2.4 Labeling line centers: In this step we create a labeled image that will share the same labels as *LLA(x,y)* and name it *LLC(x,y)*. *LLC(x,y)* will contain the line centers as where calculated *LC(x,y)* but labeled according to the text line area they traverse from *LLA(x,y)* . *LLC(x,y)* can be obtained as:

$$LLC(x,y) = LLA(x,y) * LC(x,y) \quad (8)$$

2.3 Assigning labeled components

In these steps, we use components of *LIN(x,y)* which was created in the preprocessing stage, as elementary entities to be assigned to an identified line. We create a labeled image *RES(x,y)*=0 to which we will move step by step all components from *LIN(x,y)*.

2.3.1 Assigning to line centers: In this step we select all components from *LIN(x,y)* that intersect with only one labeled line center from *LLC(x,y)* and move them from *LIN(x,y)* to *RES(x,y)* labeled as the line they intersected with in *LLC(x,y)*. Once this step is completed, *RES(x,y)* contains almost all letters, leaving in *LIN(x,y)* components that belong to two or more lines and smaller components such as accents, punctuation marks and noise. In Fig. 7 we can see components assigned at this stage as purple and *LLC(x,y)* colored as olive.

2.3.2 Assigning to line areas: In this step we apply exactly the same procedure as described in Section 2.3.1, using *LLA(x,y)* instead of *LLC(x,y)*. Once this step is completed, *RES(x,y)* contains all components that are related to only one line, leaving in *LIN(x,y)* only components that are outside all line areas, or touching components. In Fig. 7 we can see components assigned at this stage as red and *LLC(x,y)* colored as cyan.

2.3.3 Assigning remaining pixels: In this final step, we treat all components in *LIN(x,y)* as individual pixels and draw them on *RES(x,y)* as the equivalent value in *LLA(x,y)*. In detail:

$$RES'(x,y) = \begin{Bmatrix} RES(x,y) \text{ if } LIN \quad (x,y) = 0 \vee RES(x,y) \neq 0 \\ LLA(x,y) \text{ if } LIN \quad (x,y) \neq 0 \wedge RES(x,y) = 0 \end{Bmatrix} \quad (9)$$

where *RES'(x,y)* contains the final result and *RES(x,y)* is the result from Section 2.3.2. In this step, we deal with components that belonged to many lines, separating them where white path tracers passed from, and erased all pixels that where outside line areas. In Fig. 7 we can see components assigned at this stage as blue and in Fig. 8 we can see the resulting final *RES(x,y)*.

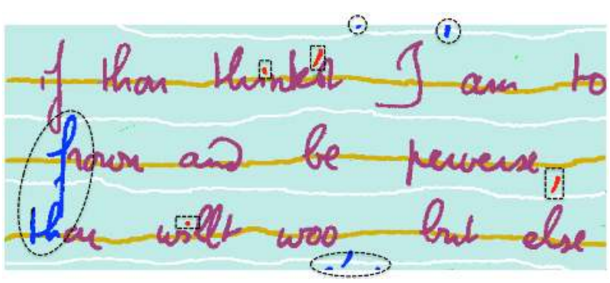


Figure 7. Component assignment: *LLA(x,y)* is shown as cyan and *LLC(x,y)* is shown as olive. In the foreground: components marked as purple where assigned in step 1, red components (in circles) in step 2 and blue components (in rectangles) in stage 3.

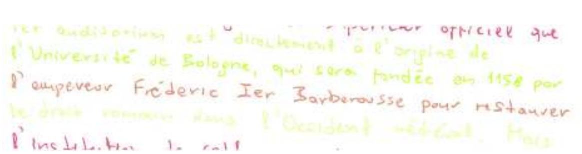


Figure 8. Part of an indicative *RES(x,y)*. White pixels are labeled as 0 (background), colored pixels are labeled according to the line they belong to.

4. Experimental results

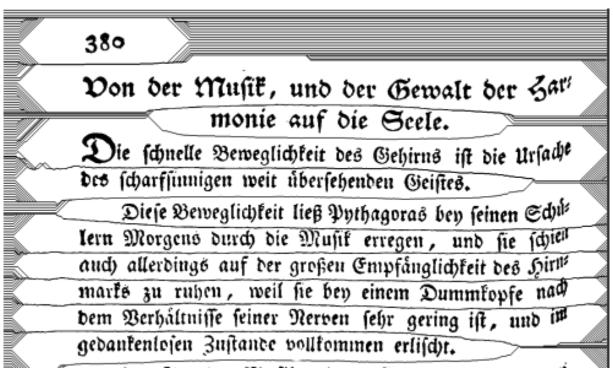
In order to estimate the efficiency of our method, we conducted experiments using the dataset from ICDAR2007 Handwriting Segmentation Contest [6], as well as a set from historical printed documents.

The dataset of ICDAR2007 Handwriting Segmentation Contest, is separated in train and test sets. The train set has 20 images containing 476 groundtruthed lines and the test set has 80 images containing 1771 groundtruthed lines. The database consists of handwritten texts, historical and contemporary, in four languages: English, French, German and Greek. The images were groundtruthed by hand. Pages contain text in one column (all lines can be extended to the images left and right borders of the image without intersecting with other lines), but text alignment varies greatly even in the same page. Images contained from 10 to more than 30 text lines. In general the database could be considered to be representative of the results of successful page segmentation in handwritten texts. We used a very simple statistical analysis of the train set to tune the coefficients of *BW* and *BH* and the minimum acceptable strip area as mentioned in Section 2.2.2. Out of 1771 groundtruthed lines in a total of 80 pages, we had 1750 one-to-one matches; 98.8 of the groundtrouthed lines. We used the same software and the same settings that were used in the competition. The performance metric is based on pixel correlation between each groundtruthed line and each identified line. We were able to compare the accuracy of our algorithm with some state-of-the-art implementations. In the experiments we measured a full implementation of our method (Table 1 row named as SHREDING), which had a performance of 98.6%. We also measured an implementation of our method up to the step of Section 2.2.2 using the intermediary image *LLA(x,y)* as the result which. In Table 1, we can observe that our method, outperformed all participant methods.

Table 1: Shredding results in handwritten text, compared with participants of ICDAR 2007 text line segmentation competition.

	Detected Lines	o2o	GT o2m	GT m2o	Detect o2m	Detect m2o	DR	RA	FM
BESUS	1904	1494	9	151	72	21	86,6%	79,7%	83,0%
DUTH-ARLSA	1894	1214	149	227	107	354	73,9%	70,2%	72,0%
ILSP-LWSeg	1773	1713	5	34	17	10	97,3%	97,0%	97,1%
PARC	1756	1604	40	76	34	85	92,2%	93,0%	92,6%
UoA-HT	1770	1674	14	54	27	28	95,5%	95,4%	95,4%
RLSA	1877	632	264	346	122	757	44,3%	45,4%	44,8%
PROJECTIONS	1892	1109	91	344	155	192	68,8%	63,2%	65,9%
LLA	1932	1521	2	33	16	4	95,2%	87,1%	90,9%
SHREDING	1782	1750	2	10	5	4	98,9%	98,3%	98,6%

In order to test the proposed methodology on historical printed documents, we made experiments with a dataset of images taken from a historical book from Eckartshausen, which was published on 1788 and is owned by the Bavarian State Library [7]. The indicative dataset consists of 38 pages from the book, 1090 lines, which where groundtruthed by hand. We used the same parameters that we extracted from the handwritten training set. In Table 2 we can see in detail the performance of our method. In order to have a point of reference we also evaluated a simple smearing method. As we can see in the results, *RES(x,y)* doesn't provide any significant improvement over *LLA(x,y)* since the occurrence of overlapping and touching components is a lot rarer. In Fig. 9, we can see an example of *LA(x,y)* on indicative printed text.


380

Von der Musik, und der Gewalt der Har-
monie auf die Seele.

Die schnelle Beweglichkeit des Gehirns ist die Ursache
des scharfsinnigen weit übersehenden Geistes.
Diese Beweglichkeit ließ Pythagoras bey seinen Schü-
lern Morgens durch die Musik erregen, und sie schien
auch allerdings auf der großen Empfänglichkeit des Hirn-
marks zu ruhen, weil sie bey einem Dummkopfe nach
dem Verhältnisse seiner Nerven sehr gering ist, und im
gedankenlosen Zustande vollkommen erlischt.

Figure 9. An indicative *LA(X,Y)* superimposed on the original printed text.

Table 2. Performance of our method of historical documents (SHREDING), compared to an intermediate state of our method (LLA) and to a simple smearing method (RLSA).

	Detected Lines	o2o	GT o2m	GT m2o	Detect o2m	Detect m2o	DR	RA	FM
RLSA	1671	1099	3	24	12	6	98,9%	66,0%	79,2%
LLA	1111	1091	11	5	3	22	98,0%	98,7%	98,3%
SHREDING	1108	1090	11	5	3	22	97,9%	98,9%	98,4%

We consider the results in segmentation of handwritten texts and historical printed documents satisfactory and encouraging.

5. Conclusions.

In the proposed methodology, we blur the image with the intention to enhance text line areas and then segment the images surface along several white paths in the blurred image. Finally we assign connected components of the original image to the appropriate line segment.

As we experimented with our method, we came to several conclusions based on the results as well as the observation of intermediary images. As can be seen in Table 1 and Table 2, *LLA(x,y)* described in Section 2.2.3 is the foundation of the results our method produces; stage 3 could be regarded as an optimization which in the case of handwritten text, gave significant improvements but in the case of printed text was totally insignificant. Our method could be expanded towards several directions among which: We could start tracers randomly in the page to deal directly with more complex layouts. We could use *LLC(x,y)* and components assigned only in the Section 2.3.1 as input for word segmentation. By observing the errors our method produced, we noticed that most of them occurred when we had great variations in letter size; we could try and define locally the blurring window depending on a more local estimation of the average letter height. Although we consider our method can deal adequately with historical printed texts, its principal goal is to deal with handwritten texts. Overall, we consider the results very encouraging and believe there is space for further research.

Acknowledgement

The research leading to these results has received funding from the European Community's Seventh Framework Programme under grant agreement n° 215064 (project IMPACT).

References

[1] F.M. Wahl, K.Y. Wong, R.G. Casey, "Block Segmentation and Text Extraction in Mixed Text/Image Documents", *Computer Graphics and Image Processing, Vol. 20*, 1982, pp. 375-390.

[2] L. Likforman-Sulem, A. Zahour, and B. Taconet, "Text Line Segmentation of Historical Documents: a Survey", *IJDAR vol 9, Springer, April 2009*, pp. 123 – 138.

[3] L. Likforman-Sulem, A. Hanimyan, C. Faure, "A Hough Based Algorithm for Extracting Text Lines in Handwritten Documents", *Proceedings of the Third International Conference on Document Analysis and Recognition*, Montreal, Canada, 1995, pp. 774-777.

[4] Y. Li, Y. Zheng, D. Doermann, and S. Jaeger,"A New Algorithm for Detecting Text Line in Handwritten Documents", *Proceedings of the Tenth International Workshop on Frontiers in Handwriting Recognition*, La Baule, Oct 2006, pp. 35 - 40.

[5] Du, X. and Pan, W. and Bui, T., "Text line segmentation in handwritten documents using Mumford-Shah model", *Pattern Recognition,* 2009, doi:10.1016/j.patcog.2008.12.021

[6] B. Gatos, A. Antonacopoulos and N. Stamatopoulos, "ICDAR2007 Handwriting Segmentation Contest", *Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR'07)*, Curitiba, Brazil, September 2007, pp. 1284-1288

[7] Carl von Eckartshausen, "Aufschlüsse zur Magie aus geprüften Erfahrungen über verborgene philosophische Wissenschaften und verdeckte Geheimnisse der Natur", Bavarian State Library, 1778.