On Influence of Line Segmentation in Efficient Word Segmentation in Old Manuscripts.

D. Fernández, J. Lladós & A. Fornés
Computer Vision Center – Dept. Ciències de la Computació
Universitat Autònoma de Barcelona
08193 Bellaterra (Cerdanyola) Barcelona, Spain
{*dfernandez, josep, afornes*}*@cvc.uab.es*

R. Manmatha
Department of Computer Science
University of Massachusetts
Amherst MA 01003, USA
manmatha@cs.umass.edu

***Abstract*—The objective of this work is to show the importance of a good line segmentation to obtain better results in the segmentation of words of historical documents. We have used the approach developed by Manmatha and Rothfeder [1] to segment words in old handwritten documents. In their work the lines of the documents are extracted using projections. In this work, we have developed an approach to segment lines more efficiently. The new line segmentation algorithm tackles with skewed, touching and noisy lines, so it is significantly improves word segmentation. Experiments using Spanish documents from the Marriages Database of the Barcelona Cathedral show that this approach reduces the error rate by more than 20%.**

***Keywords*-Segmentation; document and text processing; document analysis; handwriting analysis; heuristics; path-finding**

I. Introduction

Nowadays libraries, archives, museums, cathedrals and other organizations are filled with historical documents. The conversion of historical document collections to digital archives is of prime importance to society both in terms of information accessibility, and long-term preservation. Historical archives usually contain valuable handwritten documents. Examples are unique manuscripts written by well-known scientists, artists or writers; letters, trade forms or administrative documents kept by parish or municipalities that help to reconstruct historical sequences in a given place or time.

Handwriting recognition in historical handwritten documents is challenging. Historical documents can present physical degradation due their long lifetimes and use producing holes, stains, wrinkles, ink bleed and so on. The scanning process is difficult in historical documents. Documents are delicate and they have to be scanned with special scanners which introduce several degradations such as non-stationary noise due the illumination changes, show through effect, low contrast and, warping effect. Other kinds of difficulties are introduced by the style of the writer, for example different word shapes, curvilinear baseline due to the non-straight pen movement and touching components.

Line segmentation and word segmentation are critical early processing steps for any handwritten document recognition task. The objective is to extract word images from the documents. It is important the accuracy of this task to be high because many word recognition techniques require an accurate word segmentation. Wrongly segmented word images will cause the recognition step to fail in segmentation-based handwritten document recognition systems.

In this paper, we present a word segmentation methodology for historical handwritten documents. Most of the word segmentation approaches assume that a line segmentation approach is available and often use simple existing techniques. The objective of this work is to show the importance of having a good line segmentation to obtain better results in word segmentation. We have used the approach developed by Manmatha and Rothfeder [1] to segment words in old handwritten documents. In their work the lines of the documents are extracted using graylevel projections and word segmentation is done by filtering with anisotropic Laplacians in scale space. Essentially, they optimize for the scale by which each line should be filtered. In their work they showed that this approach for word segmentation works better than a gap metrics segmentation approach.

In this work, we have developed an approach to segment the lines more accurately. Once the lines are extracted, we have applied the method developed by Manmatha and Rothfeder to segment the words. We show that this improves the error rate by 20% compared to their original technique. Our method has as input a colour, or a gray-level, image (see Fig. 1 for a visual description of the process). The image is binarized and cleaned using an median filter. The skeleton of the image is extracted. Using the skeleton of the image, the best paths which segment the text lines are selected. Each extracted line is them segmented by applying the approach used by Manmatha and Rothfeder to segment the words. An analysis of the words obtained is applied as a postprocessing to remove false positives and to join boxes which are part of the same word.

This paper is organized as follows: Section II describes some related work on automatic word segmentation. Section III explains the method developed in this work. Section IV describes the dataset, the experiments and the results obtained. Finally, the last section concludes the paper.

978-0-7695-4774-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICFHR.2012.247

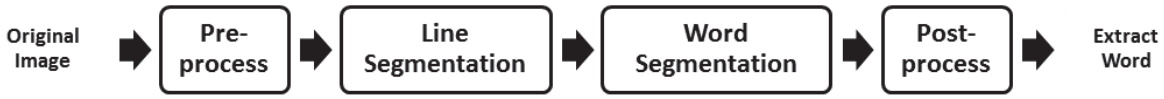


Figure 1: Illustration of the entire process.

II. Related Work

There are a wide variety of word segmentation methods reported in the literature, but they have mostly been developed for machine printed characters, where the text typically has inter-word gaps that are much larger than the inter-character gaps (gaps between characters within one word). The results obtained using these methods in handwritten documents are poor and unsatisfactory. As far as we know, the major part of the approaches are oriented to specific datasets, or are elements of integrated systems for specific tasks, for example, bank check and postal address recognition [2].

Most of the literature on handwritten document recognition is focused on modern handwritten databases. Handwritten recognition systems are often tested using specific documents created for the purpose of testing these systems and, thus, are forced in how they are created [3]. Historical handwritten documents present more drawbacks and difficulties to solve, as mentioned before. Feldbach and Tonnies [4] have proposed a bottom up method for historical church documents that needs to be set according to the type of handwriting

Most work on line segmentation in handwritten documents can be categorized in four groups. The first group makes use of the Hough transform [5]. In these methods, starting from some points of the image, the lines that fit best to these points are extracted. The second group contains the methods which use projections [6], [1]. The black pixels are projected on the vertical axis. The resulting histogram consists of regions with larger and lower concentrations of pixels. The methods of the third group use variations of smearing [7], [8]. The methods of the last group are based in dynamic programming. Liwicki et al. [9] proposed an approach for the detection of on-line handwritten text lines based on dynamic programming. They search the paths with the minimum cost between two consecutive text lines.

The literature on word segmentation shows three approaches. The first one - the gap metric approach - uses connected components (CC) to extract words. Many papers assume that each CC belongs to only one word and gaps between words are greater than the gaps between characters. Seni and Cohen [10] evaluated eight different distances measures between pairs of connected components. The second one considers the word extraction as involving determining whether each position in a text line belongs to a letter of a word or a space between two words [11]. For this task a Hidden Markov Model is used. The third approach is that of Manmatha and Rothfeder [1], which showed that their algorithm works better than gap metrics for word segmentation. This approach has been previously described.

III. Segmenting words

In this work we improve the method of Manmatha and Rothfeder [1] for word segmentation based on a more accurate line segmentation approach. In [1] an effective word segmentation method for noisy historical documents is proposed. They propose a scale-space approach where the image is first dissected into lines using a graylevel projection profiles analysis technique. The line image is filtered with an anisotropic Gaussian filter at several scales in order to produce blobs which correspond to portions of characters at small scales and to words at large scales. The appropriate scale for finding words is automatically determined by optimizing a function over scale space.

In this paper we show the influence of the line segmentation step. When text lines are straight and close to the horizontal, methods based in the analysis of projection profiles work well both for line and word segmentation. But in several cases text lines present difficulties for such methods. Such problems include skewed documents, multi-oriented lines, warping effects and so on are typical configurations. Another difficulty in handwritten documents is when lines are touching each other (descenders of one line touch ascenders of the next line). In the above cases, line segmentation based on projection profiles presents some imperfections that negatively influence the segmentation of words in the lines, and usually the resulting words are broken, or merged with other components.

In this paper we propose a word segmentation algorithm which involves replacing the line segmentation in [1] with a much better line segmentation process inspired by dynamic programming approaches. The new approach substantially improves the final accuracy. We have developed a new line segmentation process for off-line handwriting that searches a minimum energy path (Figure 1) along the medial axis which divides consecutive text lines. Our method allows to segment the lines coping with the difficulties explained before: multi-skew and touching components.

The system consists of four steps. First a pre-processing stage where the image is binarized, deskewed and the noise is removed. Afterwards lines are detected using an improved approach than in the baseline reference. Words are then

detected for each text line using a scale-space approach. The last step is a post-processing stage were false positives are deleted and over-segmentations are corrected.

A. Pre-processing

Historical manuscripts present noise and distortions due to the lifetime degradation or the scanning process. The objective of this step is to improve the quality of the applying well-known techniques.

Since the target documents are pages of books, in the scanning process black frames appear corresponding to the background. An ad-hoc process that deletes big blobs located at the margins of the image is applied. Afterwards, a median filter is used to remove noise. To reconstruct characters that are broken due to the smoothing provoked by this filter, a morphological dilation is applied. The last step of the preprocessing is the binarization of the image using Otsu's method.

B. Line Segmentation

Humans tend to write the text in blocks, and they usually use the same space between lines. In a gray level image, this characteristic can be seen as a valley, i.e. the sequence of words belonging to the same text line form a crest, and the inline space form a valley. Using this observation, we first compute the distance function on the input image finding optimal paths with minimum cost using the extracted valleys. Afterwards, we detect paths through the valleys of the document. The valleys form a skeleton of the background and consist of background points at equal distance form the words. We use a path-finding algorithm to select the best path according to a cost function.

Our approach is inspired by methods of line segmentation which use Dynamic Programming. The objective of this step is to find paths along the skeleton of the background which divide the space between two consecutive text lines. Given a set of possible initial (at left marging of the image) and end points (right margin), the algorithm links initial points with end points using a skeleton-wise following procedure. Path finding is performed by an A-star algorithm involving a cost function that is formulated in terms of the path "smoothness".

The first step of this process is to extract the skeleton of the image, from the output of the pre-processing (a binary image). An iterative thinning strategy is used. The output is a binary image which contains different paths going through the text lines and words.

Skeleton pixels are labeled as *end pixels*, *edge pixels* or *bifurcation pixels* depending on whether they have one, two or more neighbors respectively. End points are additionally labelled as *start pixels* or *end pixels* if they belong to the leftmost or rightmost columns of the image. The rest of the end pixels are considered *intermediate end pixels*. An illustration of pixel labels is shown in Figure 2.

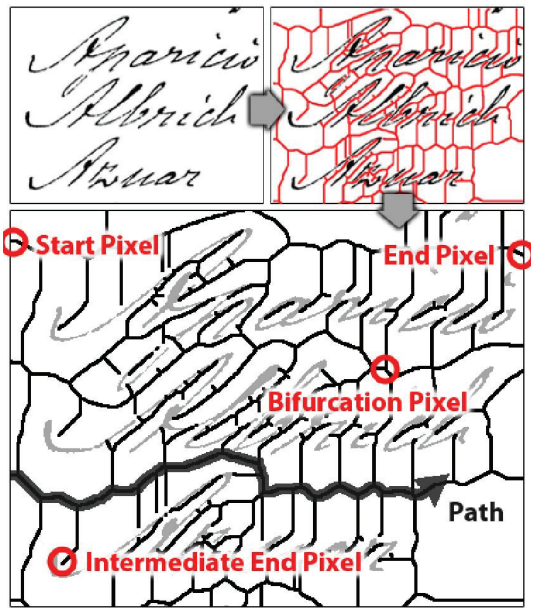


Figure 2: The skeleton is computed from the binary image. Paths, which segments text lines, are computed using the skeleton (as we can see in the bold line).

The path-finding algorithm uses the pixel labels. For each *start point* the algorithm searches a path which connects these points with an *end point* with the minimum cost. The algorithm iteratively follows connected *edge pixels* until it finds a critical point (*bifurcation pixels* or *intermediate end pixels*). If the pixel is an *intermediate end pixel*, the algorithm searches another *intermediate end pixel* close to it. We have experimentally set a radius of 100 pixels to find the closest *intermediate end pixels*. If the pixel is a *bifurcation pixel* the algorithm selects the best continuation path.

The algorithm computes a cost at each step that informally speaking measures the deviation of the path regarding the ideal one. The formulation of the cost function follows the principles of smooth continuation and connectivity. The total cost of the path is the sum of the individual costs of each step.

Formally, a path P is defined as a sequence of skeleton pixels $P = [p_1, p_2, \ldots, p_n]$. Given a point p_i the cost of the transition from p_i to the next path point p_{i+1}, denoted as $c(p_i \rightarrow p_{i+1})$ is defined as follows:

- *Null cost*: 0 if p_i is an *end pixel*, or p_i and p_{i+1} are *edge pixels* (see Equation 1).

$$C_n(p_i, p_{i+1}) = 0 \tag{1}$$

- *Bifurcation cost*: If p_i is a *bifurcation pixels* the cost is computed using the Euclidean distance between the pixel coordinates and the end of the image (see Equation 2).

$$C_b(p_i, p_{i+1}) = Euclidean(p_i, p_{i+1}) \quad (2)$$

- *Virtual Cost*: The cost between two *intermediate end pixels* is the Euclidean distance between their coordinates multiplied by the number of foreground pixels that are between them drawing a straight line between their coordinates (see Equation 3).

$$C_v(p_i, p_{i+1}) = \frac{Euclidean(p_i, p_{i+1})}{count(Pixels_{Foregr}(P_i, P_{i+1}))} \quad (3)$$

We give less weight if the path is a virtual path because the idea is to find real paths, and in the cases where a real path does not exist, or the cost is very high, to create a new one using virtual paths.

C. Word Segmentation

Once the text lines are segmented, the words of each line are extracted using the method of Manmatha and Rothfeder [1] and we briefly summarize the approach used. The input is a gray-scale image, then, for each line, it is converted to gray-scale and used as input to the algorithm of word segmentation. The line image is then filtered with an anistropic Laplacian filter. The filtered output is thresholded to create a set of blobs. At a certain scale the blobs are more likely to correspond to words. This scale is automatically found by doing an optimization over scale space. The blobs at this optimum scale correspond to the words.

D. Post-processing

The output of the binary image obtained from the preprocessing step contains a set of connected components (CC) from each line. Assume that box b_j contains CC $\{a_1, a_2, \ldots, a_m\}$, where a_j represents both the CC and its area. We consider the box b_j its a valid box if

$$\frac{max\{a_1, a_2, ..., a_m\}}{a_j} > 0.1 \quad (4)$$

and

$$0.99 > \frac{width(b_j)}{heigth(b_j)} > 0.1 \quad (5)$$

To solve the problem of over-segmentation, two boxes (b_k, b_h) on the same line which overlap by more than 10%, assume that the box b_k contains CC $\{a_{k1}, a_{k2}, \ldots, a_{kn}\}$ and the box b_h contains CC $\{a_{h1}, a_{h2}, \ldots, a_{hn}\}$. We consider that both boxes allow to the same word if

$$max\{a_{k1}, a_{k2}, \ldots, a_{kn}\} == max\{a_{h1}, a_{h2}, \ldots, a_{hn}\} \quad (6)$$

This says that the largest area of b_k must be the same of the largest area of b_h.

IV. Experiments and results

A. Marriage Licenses Databases

We have applied our work in a demography application. In particular, word spotting is applied to the manuscripts called *Llibre d'Esposalles*, a set of books written between 1451 and 1905. This corpus records marriage and the corresponding fees paid according to the social status of the families. It is conserved at the Archives of the Barcelona Cathedral and comprises 244 books with information on approximately 550.000 marriages celebrated in over 250 parishes. Each book contains the marriages of two years, and was written by a different writer. Information extraction from these manuscripts is of key relevance for scholars in social sciences to study the demographical changes over five centuries. In this work we have used 10 pages of the *volume 232* to do the experiments.

B. Experiments

The algorithm has been developed and tested on 10 documents from the *volume 232* of the archive of the Cathedral of Barcelona. To reduce the runtime, the images have been reduced to a 20% of their original size.

We have done three experiments. The first one apply our new approach of optimized line segmentation instead the projection based method that Manmatha and Rothfeder use in their work. The second experiment uses the original algorithm of Manmatha and Rothfeder over our documents. The last experiment is similar to the first one but without post-processing. Tables I, II and III show, respectively, the results for each of the experiments.

In the tables, true positives are those for which a bounding box is generated for a real word. Missed words are those for which no bounding box is generated. Over segmentation occurs when two or more bounding boxes are generated for one word. Under-segmentation occurs when two or more words lie within one bounding box. Since the ground truth is generated on a per word basis, three words in a box count as three errors. Extra boxes are wrong boxes or false positives. The total errors column is the sum of errors in the other columns.

C. Results

Using a post-process to remove extra boxes and over segmentation, we obtain better results. The number of words increase from 73% to 79%, the over segmentation is reduced, under segmentation is similar and extra boxes are reduced drastically from 9% to 0.37%. Instead, missing words increase when we apply post-processing. Applying post-processing we discard mostly extra boxes produced by the margins and noise in the document which may be detected as words. But, in some cases, the post-processing detects bounding boxes which are overlapping, as part of the same word, and they are joined.

Comparing our new approach with the work developed by Manmatha and Rothfeder with the same images, we observe a clear improvement. True positives increase from 52% to 79%, missed words decrease from 9% to 4%, over segmentation decrease drastically from 26% to 0.5% and extra words are also reduced. But the under segmentation increases using our method from 9% to 14%, because the words are close and, even for humans it is difficult to separate. Note that if a word is divided into two boxes it counts as two under segmentation errors. This explains the large number of under segmentation errors.

Figure 3b and 3a show the results using our optimized line segmentation and the Manmatha and Rothfeders method. We have used bounding boxes to show the results because second method uses bounding boxes. Although, the accuracy of the word segmentation is higher using an optimized line segmentation (3c).

Doc.	TP(%)	MW(%)	OS(%)	US(%)	EB(%)	TE(%)
1	73.46	3.85	0.77	21.92	0.00	26.54
2	85.23	3.41	0.00	10.98	0.38	14.77
3	85.23	2.65	0.38	10.98	0.76	14.77
4	73.90	6.25	0.37	19.49	0.00	26.10
5	75.55	5.47	0.36	18.25	0.36	24.45
6	74.34	1.51	1.89	21.51	0.75	25.66
7	73.90	6.62	0.00	18.01	1.47	26.10
8	85.13	4.46	0.37	10.04	0.00	14.87
9	82.77	7.49	0.37	9.36	0.00	17.23
10	83.97	6.11	0.76	9.16	0.00	16.03
Average	**79.35**	**4.78**	**0.53**	**14.97**	**0.37**	**20.65**

Table I: Results combining an optimized line segmentation with Manmatha and Rothmeder's method. The metrics used are: True Positives (TP), Missed Words (MW), Over Segmentation (OS), Under Segmentation (US), Extra Boxes (EB) and Total Errors (TE).

Doc.	TP(%)	MW(%)	OS(%)	US(%)	EB(%)	TE(%)
1	54.75	11.79	25.10	5.32	3.04	45.25
2	49.25	12.31	28.73	7.09	2.61	50.75
3	56.44	2.65	25.38	14.77	0.76	43.56
4	60.22	8.76	19.71	10.22	1.09	39.78
5	49.65	13.12	30.50	2.84	3.90	50.35
6	44.53	12.41	35.04	3.65	4.38	55.47
7	43.75	8.82	30.88	15.07	1.47	56.25
8	51.67	6.32	22.30	18.59	1.12	48.33
9	60.30	10.86	16.48	11.99	0.37	39.70
10	53.38	10.90	27.07	6.77	1.88	46.62
Average	**52.39**	**9.79**	**26.12**	**9.63**	**2.06**	**47.61**

Table II: Results of Manmatha and Rothmeder's method.

Doc.	TP(%)	MW(%)	OS(%)	US(%)	EB(%)	TE(%)
1	63.40	1.31	0.65	20.26	14.38	36.60
2	80.22	2.16	0.00	12.95	4.68	19.78
3	75.08	1.66	0.33	10.96	11.96	24.92
4	69.90	2.68	0.00	18.06	9.36	30.10
5	69.57	3.01	1.00	17.39	9.03	30.43
6	66.67	1.06	2.13	22.70	7.45	33.33
7	73.94	1.41	1.06	17.96	5.63	26.06
8	78.26	0.67	1.67	9.36	10.03	21.74
9	77.10	2.36	0.67	10.44	9.43	22.90
10	79.51	2.08	0.69	8.68	9.03	20.49
Average	**73.37**	**1.84**	**0.82**	**14.88**	**9.10**	**26.63**

Table III: Results combining an optimized line segmentation with Manmatha and Rothmeder's method (without post-processing).

V. Conclusion

We have presented a new approach of word segmentation in historical documents. Our approach replaces the line segmentation process developed in the work of Manmatha and Rothfeder [1] by a new optimized line segmentation approach. The algorithm uses adaptive paths to segment the lines.

We have evaluated our new approach using a data set selected form the archive of the Cathedral of Barcelona. This archive has 244 volumes with hundreds of pages each one. We have selected 10 representative pages of the volume 232 to do the experiments of this work.

The objective of this work, to show the improvement of the results using optimized line segmentation, has been met. Our method obtains better results than the method of Manmatha and Rothfeder over old handwritten documents due the lines are segmented in a best way.

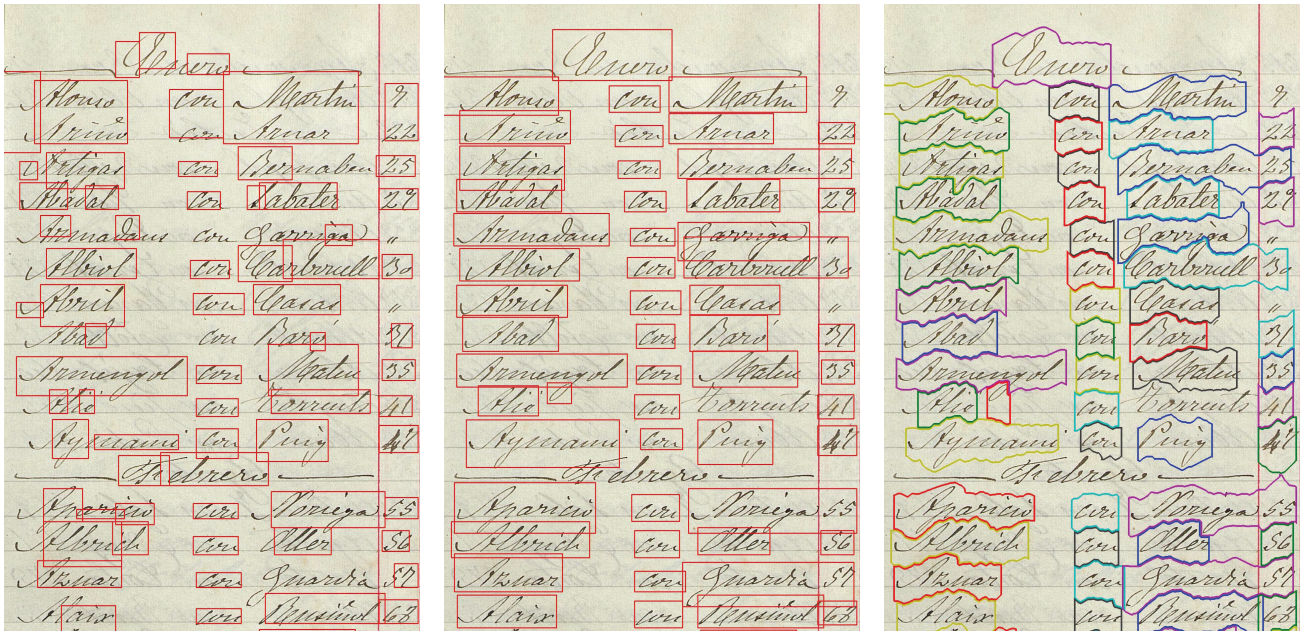
Acknowledgment

The authors thank the *CED-UAB* and the Cathedral of Barcelona for providing the images. D. Fernandez, J. Llados and A. Fornes are partially supported by the Spanish projects TIN2011-24631, TIN2009-14633-C03-03 and CSD2007-00018, by the EU project ERC-2010-AdG-20100407-269796 and by a research grant of the UAB (471-01-8/09). R. Manmatha is supported by the Center for Intelligent Information Retrieval and by NSF grant #IIS-0910884. Any opinions, findings and conclusions or recommendations expressed in this material are the authors and do not necessarily reflect those of the sponsor.

References

[1] R. Manmatha and J. Rothfeder, "A scale space approach for automatically segmenting words from historical handwritten documents," *I. trans. PAMI*, vol. 27, pp. 1212–1225, 2005.

[2] G. Nagy, "Twenty years of document image analysis in pami," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 1, pp. 38 – 62, jan 2000.



(a) (b) (c)

Figure 3: Results using a projection base method to segment lines (3a) and results using an optimized line segmentation (3b). In both methods the words are extracted using the approach developed by the Manmatha and Rothfeder's. In image 3c observe the accuracy of the optimized line segmentation, instead of using bounding boxes.

[3] U. Marti and H. Bunke, "A full English sentence database for off-line handwriting recognition," in *I. conf. ICDAR*. IEEE, 1999, pp. 705–708.

[4] M. Feldbach and K. Tonnies, "Line detection and segmentation in historical church registers," in *I. conf. ICDAR*. IEEE, 2001, pp. 743–747.

[5] G. Louloudis, B. Gatos, and I. Pratikakis, "A block-based Hough transform mapping for text line detection in handwritten documents," *10th I. Workshop on Frontiers in Handwriting Recognition*, 2006.

[6] M. Arivazhagan, H. Srinivasan, and S. Srihari, "A statistical approach to line segmentation in handwritten documents," in *Proc. of SPIE*, vol. 6500, 2007, pp. 6500T–1–11.

[7] Z. Shi and V. Govindaraju, "Line separation for complex document images using fuzzy runlength," in *Doc. Image Analysis for Libraries*. IEEE, 2004, pp. 306–312.

[8] Z. Razak, K. Zulkiflee, M. Yamani, I. Idris, E. M. Tamil, M. Noorzaily, M. Noor, R. Salleh, M. Yaacob, and Z. M. Yusof, "Off-line handwriting text line segmentation: a review," *IJCSNS*, vol. 8, no. 7, pp. 12–20, 2008.

[9] M. Liwicki, E. Indermuhle, and H. Bunke, "On-Line Handwritten Text Line Detection Using Dynamic Programming," *I. conf. ICDAR*, pp. 447–451, Sep. 2007.

[10] G. Seni and E. Cohen, "External word segmentation of off-line handwritten text lines," *Pattern Recognition*, vol. 27, pp. 41–52, 1994.

[11] F. Luthy and T. Varga, "Using hidden Markov models as a tool for handwritten text line segmentation," *I. conf. ICDAR*, pp. 8–8, 2007.