# Robust Method for the Text Line Detection and Splitting of Overlapping Text in the Latin Manuscripts

Jakub Leszek Pach, Piotr Bilski
[1]*Institute of Radioelectronics, Warsaw University of Technology, Poland*
{*jakub.pach,pbilski*}*@ire.pw.edu.pl*

**Abstract.** The paper presents the modified method of the text lines separation in the handwritten manuscripts. Such an approach is required for the medieval text analysis, where multiple text lines overlap and are written at different angles. The proposed approach consists in dividing the bounding boxes into smaller components based on the points of the character curves intersection. The method considers the askew text lines, producing non-rectangular zones between the neighboring lines.

**Key words:** document analysis, unconstrained handwriting, Hough transform, text line detection, connected component analysis, histogram analysis.

## 1. Introduction

The manuscript author identification is the complex and time consuming task, requiring multiple steps, which include the scanned document preprocessing, extracting the handwriting features and, based on them, making decision. The first stage is crucial for the task, as the classification efficiency strongly depends on the ability to identify features maximizing the separation of manuscripts belonging to different authors. Before the classification can be executed, multiple image processing operations (such as the text line detection or word separation) must be commenced. The difficulty of these tasks depends on the character of the manuscript, its origin or the time it was created. For instance, letters in the medieval Latin manuscripts are extremely dense, with multiple overlapping letters from the neighboring text lines. Also, the text lines may be straight (parallel to each other) or aslant at various angles. This abundance of problems allows for classifying the manuscripts regarding their difficulty for the image processing operations. Existing approaches for each stage are grouped according to this taxonomy. In most papers, simpler texts are processed, leaving open field for the analysis of the complex documents.

In the following paper, novel methods for two operations performed in the first stage of the image processing are presented. They include the text line detection and division of overlapping text blocks into separate sections. Both are important for the subsequent segmentation, i.e. extracting separate words from text fragments. Although advanced mathematical transformation methods are used for both tasks, the more complex documents require the new approach characterized by a higher accuracy than before. The

proposed algorithms are verified against selected Latin documents collected from Polish National Library.

The paper structure is as follows. In Section 2 the existing approaches are categorized and discussed. In section 3 the generic word segmentation system is presented, including subsequent stages. Section 4 introduces the novel algorithms implemented to solve the presented problems. In Section 5 results of the manuscripts' processing are presented. Section 6 contains conclusions and future prospects for the proposed approaches.

## 2. State of the art

To solve the considered tasks, i.e. the text line detection and the word fragments separation, various algorithms were used in the past. Characteristics of the most popular approaches are presented below to show advantages and drawbacks of the existing methods.

### 2.1. Text line detection

The text line detection is well established in the field of historical documents processing. In general, the developed algorithms fall into the following groups, which will be briefly explained below:

- Profile projection
- Hough transform-based
- Exploiting fuzzy run length.

The profile projection approach uses the histogram analysis to detect rows of densely grouped pixels [4]. Each row is one stripe of the histogram with the value proportional to the number of black pixels (composing words and letters). This way the pattern consisting of peaks or valleys is generated, where the former indicate the text lines. In [2] the initial image was additionally partitioned into vertical segments. For each segment, the horizontal histogram was calculated, leading to the higher line detection accuracy. Histogram-based methods have difficulties in detecting askew lines, therefore their modifications are required. It is assumed that the subsequent text lines in the original image are almost parallel to each other, which is not true in general.

Other approaches ([1, 9]) use the Hough transform which is able to locate aslant lines of text. The original pixel (in the Cartesian space) is represented as the trajectory in the polar space. The intersections of trajectories identify the most probable text lines. The probability of the correct line identification in the particular point is proportional to the number of intersecting trajectories in this location. The original approach was optimized to decrease the computational complexity [9]. Modifications are proposed to increase the accuracy of the line detection [14].

The fuzzy run length method exploits the effect of smudging the image along the horizontal axis [6]. This way the original words and sentences become "fuzzy" (with smudged contours proportional to the number of black pixels in the continuous block constructing the letter or word). The line detection is then executed locating three lines, i.e. the top and bottom border, and the middle between them, which identifies the actual text line. The approach is flexible, computationally saving and is able to detect leaning lines at the angle from the range of $\pm 45°$. It fails to detect curved (parabolic shaped) lines. The algorithm was tested on the official historical manuscripts.

Besides the groups of presented approaches, less popular solutions exist. For instance, the shortest spanning tree was used to analyze the Arabic text [7]. The clustering of CCs may also be useful, as was presented in [3]. In [8] the Adaptive Local Connectivity Map was used for this purpose. These methods are not yet fully explored, therefore their computational demands and detection accuracy should be optimized.

### 2.2. Extraction of overlapping text blocks from detected lines

The second stage is executed for all CCs, intersected by at least two lines. To extract the particular words (for instance, having the common part), the skeleton must be generated, i.e. the thin shape being the center of the monochrome area. As the original text is binarized, only black and white colors remain. Depending on which color is processed, the foreground or background skeleton is generated. As the result, character or background contours are constructed, respectively. The ideas behind these approaches are as follows [5, 12]:

- Foreground processing, i.e. working on the black pixels over the white background (where black color is represented by the value 0, while white is represented by 1). They are applied to the text with overlapping characters from different blocks and generate contours of characters.
- Background processing, i.e. working on the white pixels from the binary text image. The background skeleton of empty spaces surrounding characters is generated here.
- Recognition of single characters, separating them from the text line.

Existing approaches fail in distinguishing between overlapping characters in the aslant text lines (where letters overlap under different angles), which is the typical in Latin manuscripts. The method proposed in the paper solves the presented problems ensuring higher accuracies.

## 3. Text processing architecture

This section presents the generic architecture for the image processing during the word separation procedure. Operations executed here prepare the separate words or their fragments to the features extraction (such as the letter thickness or convexity factor),

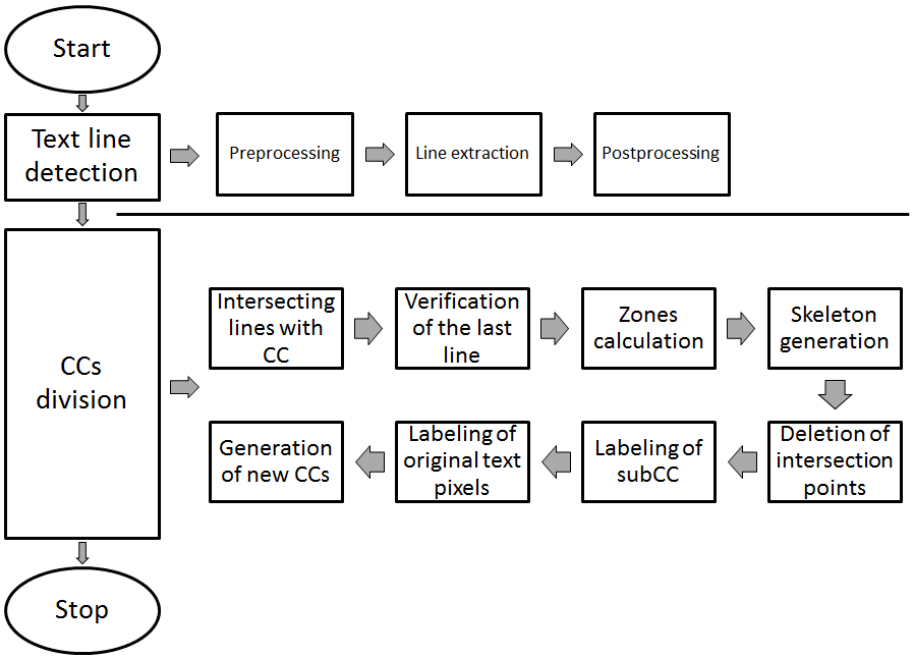


Fig. 1. The generic architecture of the word segmentation system

which further will be used to identify the author. In Fig. 1 two main operations (i.e. the text line detection and division of CCs into separate fragments) are presented. Each contains some smaller steps, executed sequentially. The text line detection consists in extracting fragments of the binarized image, which belong to the same line. This is the first step to extract dense fragments of the text, further used to identify the author. This operation was considered previously, therefore it will be presented shortly, while the CCs division requires more attention. Its main steps are extracting zones (areas belonging to the particular text line), creating skeletons of the writing curves and dividing CCs containing the text from two intersecting lines into separate areas (subCCs). In each subCC, pixels of the handwriting are added to the nearest skeleton and labeled as the dense fragment. Details of these operations are in Section 4.2.

The fundamental structure for the text processing is the Connected Component (CC). It is the dense set of the pixels marked by ink (handwriting) which borders are defined by the bounding box. Before CCs can be extracted, a list of operations must be performed in the preprocessing stage. Every document is first scaled down to the minimum acceptable resolution. The historical manuscripts are usually scanned with very high

resolutions, exceeding the algorithm's requirements. To shorten the processing duration, their resolution is decreased. Next, the binarization is performed to obtain only black and white image, where the white color represents the text, while the black is the background. The CCs are then identified and separated into three groups:

- $S_1$, containing the "short" characters, spanning over the single text line. They meet the following requirement:

$$cc_i \in S1 \Leftrightarrow \left\{ \left( \frac{1}{2} \cdot \bar{h} < h_i < 3 \cdot \bar{h} \right) \wedge \left( w_i > \frac{3}{2} \cdot \bar{w} \right) \right\} , \tag{1}$$

  where $h_i$ is the height of the $i$-th CC, and $w_i$ its width, and $\bar{h}$ and $\bar{w}$ are their mean values, respectively.

- $S_2$, containing all characters spanning over multiple text lines. This includes capital letters and the ones with long tails, overlapping the neighboring text lines:

$$cc_i \in S2 \Leftrightarrow \left( h_i >= 3 \cdot \bar{h} \right) . \tag{2}$$

- $S_3$, containing remaining elements of the text, such as accents, dots, etc.:

$$cc_i \in S3 \Leftrightarrow \left( \frac{1}{2} \cdot \bar{h} > h_i \right) . \tag{3}$$

The extracted CCs are processed by the Hough transform to find candidates for the text lines. Every CC is divided into the selected number of blocks. In each block the central point is calculated with the coordinates of the ink pixel being the closest one to the geometric center of the image fragment. For each point the accumulator array is generated, illustrating the position of the particular text lines intersecting the subsequent central pixels. The text line is detected if pixels from at least nine blocks intersect (which means the same straight line goes through all of them). From the remaining elements of the accumulator array the additional text lines are identified if they are parallel to the ones detected previously. In the postprocessing stage the redundant and "false" lines are deleted and the ones constructed by CCs not being the part of any line are added.

Separation of the intersecting characters from the neighboring lines requires the preceding extraction of skeletons of the handwriting (i.e. the thin shapes – one-pixel broad – of the written characters), which are the core elements of the text. Their identification allows for division of CCs containing shapes from neighboring text lines (vertically) into smaller fragments, each containing the text from one line only. The problem is the assignment of the original text pixels to the particular skeletons, especially in the case of intersecting fragments.

## 4. Description of the implemented algorithms

To correctly separate fragments of the text into single words or phrases (consisting of continuous curves created by the ink) two operations are performed, presented below in details.

### 4.1. Text line detection

This step consists in the standard sequence of operations, as explained above. The proposed method was already implemented in the previous research [14]. Its main idea is the introduction of the supporting CCs to increase the number of detected lines. After the binarization using the Otsu algorithm and generation of three groups of CCs, only the group $S_1$ is considered for the text lines detection. We introduced the supporting CCs, which are built based on histograms calculated horizontally on the vertical fragments of the text. They were added to the set $S_1$ and processed similarly as already existing CCs.

Calculating the Hough transform for the extended set of CCs allowed for increasing the number of correctly detected text lines. Finally, the incorrectly detected and repeating lines are eliminated to present the unique set.

### 4.2. Text fragmentation

After detecting text lines, it is possible to have CCs intersected by multiple (at least two) lines. The following procedure is proposed by us to separate such a complex CC into separate ones (along the y axis) if it contains multiple text lines.

1. Zones $Z_i$ are created as the areas between the neighboring lines (the $i$-th zone is created between the lines $i$ and $i$+1). In contrast to [13], where all lines were constructed as parallel to each other and to the bounding box, we assume the angle between the neighboring lines may be between $-5^\circ$ and $+5^\circ$ (while the angle between the parallel lines is $0^\circ$). Therefore for each zone candidate it is checked if the line is parallel or not. In the first case, the zone is defined as the rectangle limited by the intersection points between the neighboring lines and boundaries of the CC [9] (Fig. 2). If the lines are not parallel, the proposed procedure is more computationally complex. It constructs the zone area by determining the top and the bottom pixel of the zone for each column of the image (each point in the $x$ dimension). The proposed approach is more accurate than existing ones [9]; the lines are considered non-parallel even if the difference between the vertical positions of extreme line points belonging to the CC is 1. Two main cases are considered here. The actual text lines may form the parallelogram or the trapezoid (Fig. 3). The result of both effects may be the incorrect assignment of the handwriting pixels to the particular skeleton, which is illustrated in Fig. 4. Here the problem may be separating text pixels from the second and the third line. The letters "e" from the former and "l" (the fragment indicated by the black circle) from the latter should be

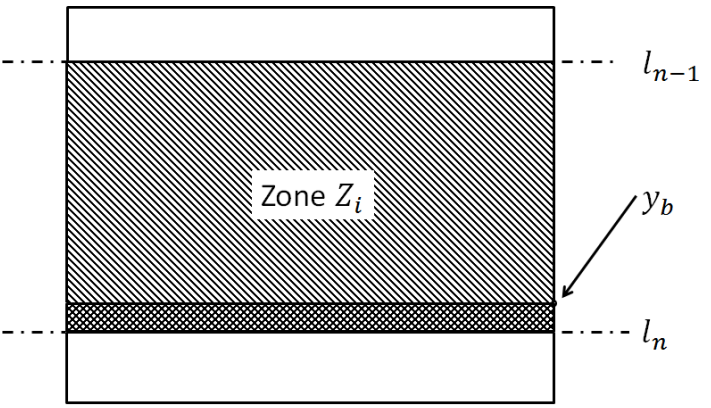


Fig. 2. The zone $Z_i$ with the 10% of the area above the text line, where the condition (4) is met.

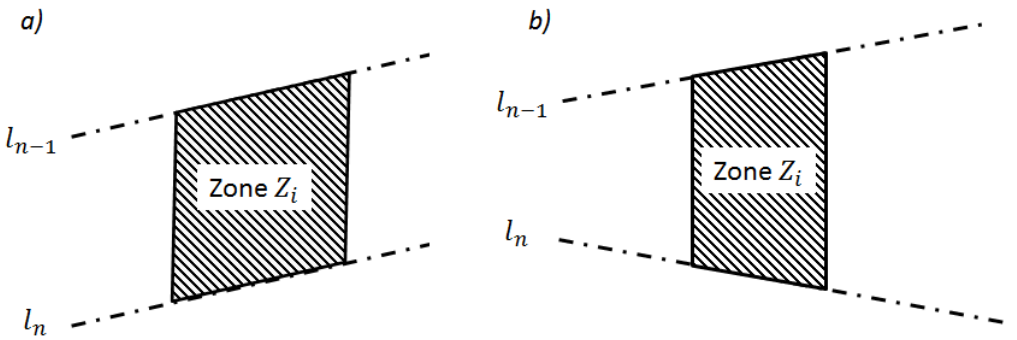


Fig. 3. Examples of the non-rectangular zones, created by the neighboring text lines: parallelogram (a) and trapezoid (b).

separated. The method from [13] will cut the text according to the thick dotted lines, while our approach will be more precise, separating both letters along the aslant grayed lines.

2. In each CC it is checked if the last line from the bottom is indeed the text line. In the Latin manuscript some letters (such as "j", "y" or "f") may be written with long strokes. In such a case, the multiple set of strokes at the same vertical level may be detected by the previously presented procedure as the separate text line. Normally such long fragments of letters overlap with the characters from the lower text line, but in the sequence of words (after which there is the significant empty vertical space), they

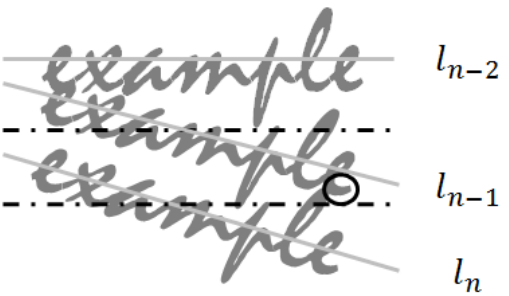


Fig. 4. Example of the neighboring text lines non-parallel to each other and the subsequent assignment of pixels of the intersecting characters to particular skeletons.

can be detected as the separate line. To avoid this false detection, 10% of the last zone (between the preceding and the bottom line) is processed to determine if the fraction of the black points (ink traces) here is at least 8% of the whole area:

$$0.08 < \frac{\lfloor Z_i \rfloor_{0.1}}{Z_i} = \frac{\sum_{x_b y_b}^{x_{n-1} y_{n-1}} I_{x,y}}{\sum_{x_s y_s}^{x_n y_n} I_{x,y}}\,, \tag{4}$$

$$y_b = (y_s - y_n) \cdot 10\,. \tag{5}$$

The last lines in the CC not meeting this criterion are eliminated as they do not construct the actual text.

3. Skeletons of the characters are created in each CC. This is the method of deciding, which parts of the overlapping curves belong to the particular text line. The skeleton is constructed as the one-pixel width shape going through the center of the character curve (which, if the document was scanned with the resolution high enough, will be much thicker). After this operation, the character shapes with minimum thickness are obtained. Next, the points of intersection between the characters belonging to different text lines are found and eliminated. The intersection point contains the ink pixel in the center and at least three neighboring black pixels. Finally, separated curves are assigned to the neighboring text lines. The fragments laying on the text lines are just assigned to them. For the fragments not laying on the text line, the subCC (rectangles constructed on the boundaries of the curve) are created. They are assigned to the text line being the closest one (in the Euclidean sense) to them. If there is no intersection in the CC, the zone $Z_i$ is divided into two halves and black pixels from both areas are assigned to the corresponding text lines. Each skeleton point belonging to the same text line is labeled with this line's identifier.

4. The original character pixels are labeled according to the smallest distance from

the skeleton pixels. As the result, the whole text is assigned to the particular text line identifiers, illustrated in the following pictures with the subsequent colors.

## 5. Experimental results

The results of processing the document [15] (containing hundreds of pages) with our approach are shown in Fig. 5. Two pages of 25 selected for experiments are presented here. On the left side, the original image is presented. The right side contains results of computations. Each detected line with the text belonging to it is indicated by one of three colors: blue, red and green (used interchangeably to show, which elements of the text were assigned to which line). The black color indicates the text fragments not assigned to any line, which is not considered error, but used as a fail safe. If the bounding box contains the character too far from the text line, it is not assigned to it, although might be, if additional requirements (introduced during the future research) are met. In the ideal case, the whole line should be identified by the separate color. As can be seen, some fragments of the the lines are described by different colors, which indicates the error of identification (for instance, the left part of the third and the fourth line from the top of the first page in Fig. 5). This illustrates the complexity of the identification problem, especially if the text lines are not parallel (in the first page the text forms the parabolic shapes instead of the straight lines).

In the presented case, the proposed methodology gives acceptable results, as the line detection accuracy is above 90%. The segmentation accuracy for the correctly detected lines is difficult to assess at this stage - the appropriate assessment method will be proposed in the next paper. Some problems with the text segmentation are caused by the previous step, i.e. the text line detection, which should be improved in the future. For instance, the non-adaptive Otsu algorithm should be replaced with a more robust approach. Also, is was observed that the size of the $S_1$ CC set influences the lines identification and text segmentation accuracy.

## 6. Conclusions

The method presented in the paper ensures the more accurate division of words in the manuscript than the approaches used so far. The well established algorithm [9] assumes the text lines are in the horizontal position with small parts of the overlapping text. The presented problem is more difficult, as virtually all lines are interconnected with each other. This makes generation of the CCs a complex process, as many blocks will be intersected by the multiple lines. To avoid errors in the subsequent handwriting processing stages, the number of CCs assigned to the correct text lines should be maximized. The proposed approach is useful in dividing the bounding box into smaller counterparts even if the character curves intensely overlap. The proposed trapezoidal shape of the

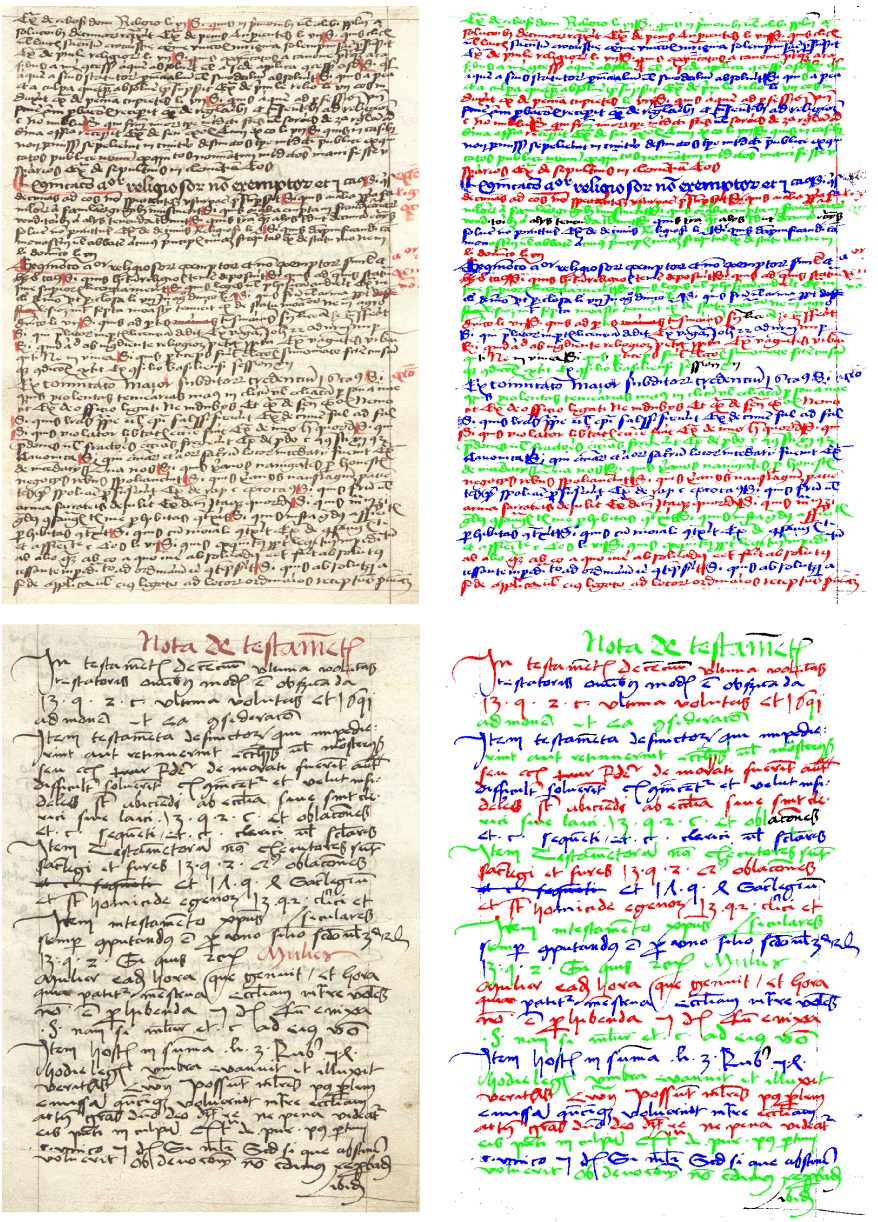

Fig. 5. Illustration of the performance of the proposed method. The left side presents the original text, while the right side contains the results of the processing.

bounding box produces more accurate zone areas. This allows for the better assignment of the particular CC fragments to the corresponding text lines. Such small differences between the obtained zones may have impact on the next stage, i.e. the handwriting features extraction. In shorter CCs the chance of having parallel lines is higher than that for the long ones.

## References

**1988**

[1] L. A. Fletcher and R. Kasturi : A robust algorithm for text string separation from mixed text/graphics images. In IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 10, no. 6, pp. 910-918.

**1995**

[2] L. Likforman-Sulem and A. Hanimyan and C. Faure : A Hough based algorithm for extracting text lines in handwritten documents. In Proceedings of the Third International Conference on Document Analysis and Recognition, vol. 2, pp. 774-771.

[3] I. S. I. Abuhaiba, S. Datta, and M. J. J. Holt : Line extraction and stroke ordering of text pages. In the Third International Conference on Document Analysis and Recognition, vol. 1, pp. 390-393.

**1999**

[4] E. Bruzzone and M. C. Coffetti : An algorithm for extracting cursive text lines. In Proceedings of the Fifth International Conference on Document Analysis and Recognition ICDAR '99, pp. 749-752.

**2000**

[5] Yi-Kai Chen and Jhing-Fa Wang : Segmentation of single- or multiple-touching handwritten numeral string using background and foreground analysis. In IEEE Transactions on Pattern Analysis and Machine Intelligence 2000, pp. 1304-1317.

**2004**

[6] Z. Shi and V. Govindaraju : Line separation for complex document images using fuzzy runlength. In First International Workshop on Document Image Analysis for Libraries, pp. 306-312.

[7] S. Nicolas, T. Paquet, and L. Heutte : Text line segmentation in handwritten document using a production system. In IWFHR-9 2004. Ninth International Workshop on Frontiers in Handwriting Recognition, pp. 245-250.

**2005**

[8] Z. Shi, S. Setlur, and V. Govindaraju : Text extraction from gray scale historical document images using adaptive local connectivity map. In Eighth International Conference on Document Analysis and Recognition, vol. 2, pp. 794-798.

**2006**

[9] G. Louloudis, B. Gatos, I. Pratikakis, and K. Halatsis : A block-based Hough transform mapping for text line detection in handwritten documents. In Tenth International Workshop on Frontiers in Handwriting Recognition.

**2007**

[10] M. Arivazhagan, H. Srinivasan, and S. Srihari: A statistical approach to line segmentation in handwritten documents. In Proc. SPIE 6500, Document Recognition and Retrieval XIV, 65000T.

**2011**

[11] A. Alaei, P. Nagabhushan, and U. Pal: A new text-line alignment approach based on piece-wise painting algorithm for handwritten documents. In International Conference on Document Analysis and Recognition (ICDAR), pp. 324-328.

**2014**

[12] Chamchong, R. and Chun Che Fung: A combined method of segmentation for connected handwritten on palm leaf manuscripts. In 2014 IEEE International Conference on Systems, Man and Cybernetics (SMC), pp. 4158-4161.

[13] Gatos, B. and Louloudis, G. and Stamatopoulos, N.: Segmentation of Historical Handwritten Documents into Text Zones and Text Lines. In 14th International Conference on Frontiers in Handwriting Recognition (ICFHR 2014), pp. 464-469.

**2015**

[14] J. L. Pach, P. Bilski: A Robust Text Line Detection in Complex Handwritten Documents. In 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. (IDAACs 2015), 24-26 Sept. 2015, Warsaw, Poland, pp. 271-275.

[15] Miscellanea theologica 2015. [Online]. Available: `http://polona.pl/item/12909419/0/`.