computing@computingonline.net
www.computingonline.net

Print ISSN 1727-6209
On-line ISSN 2312-5381
International Journal of Computing

A ROBUST BINARIZATION AND TEXT LINE DETECTION IN HISTORICAL HANDWRITTEN DOCUMENTS ANALYSIS

Jakub Leszek Pach, Piotr Bilski

Institute of Radioelectronics and Multimedia Technologies, Warsaw University of Technology,
Warsaw 00-665, Poland, jakub.pach@ire.pw.edu.pl, pbilski@ire.pw.edu.pl

Abstract: In this paper, we present a novel method of detecting text lines in handwritten documents based on the Block-Based Hough Transform. To maximize its efficiency, the robust binarization algorithm was applied. It is based on the Gaussian filtering and tackles the non-uniform luminance. The proposed technique consists of three steps: preprocessing, detecting of potential text lines and eliminating the false ones. The first step covers the image binarization, extraction of connected components and selection of supporting connected components based on the local maxima in the vertical histogram stripes. Secondly, the appropriate subset of connected components supplemented by one-point components is selected. Finally, the block-based Hough transform is applied to detect potential text lines and found the ones identified incorrectly. The proposed method is applied to the analysis of the fifteenth century Latin manuscripts. Our approach is more effective than the traditional ones, in the best cases by twenty percent.

Keywords: Document analysis, unconstrained handwriting, Hough transform, text line detection.

1. INTRODUCTION

The analysis of handwriting recognition is mainly used to extract content, but also gives the opportunity to determine characteristics of individual letters. This domain (being the combination of historical and computer sciences) can be used for the paleography or as judicial forensics to identify the author of the document. Detection of text lines is the first step in the manuscript analysis. Errors arising at this point create the noise, degrading the efficiency of the methods used at later stages (such as the text segmentation). The most popular approaches implemented for this task first extract the rectangular areas surrounding the separable groups of letters (Connected Components – CC, also called bounding boxes) from the scanned text image and process them by the block-based Hough transform mapping [1] to detect the maximum number of text lines. The number of CCs in the most difficult documents is usually too small to detect all text lines correctly. Fragments of letters often overlap, making their separation difficult. Also, lines in the typical manuscript are not necessarily parallel (because each is written at the different angle). A good example of this problem is the 15th century text written using the Latin alphabet.

In this paper we propose a modified approach to the lines detection task, by adding the group of Supporting CCs (SCC) to the initially obtained set of CCs. This allows for detecting more lines than in previously existing methods. Then, the extended Hough transform is used to correctly identify more text lines than other available methods. The paper is organized as follows: in Section 2, the problem to solve is defined and related work discussed. In Section 3, the method of segmenting text lines is introduced in detail. In Section 4 the experimental results are presented, while Section 5 contains conclusions and future prospects.

2. PROBLEM STATEMENT AND RELATED WORK

Text lines detection is the key operation in the complex manuscript author identification system, which processes features of handwriting. To achieve this aim, three steps are required:

- preprocessing,
- feature extraction and dictionary construction,
- author's identification.

In the preprocessing stage the line detection is implemented. Because the archival manuscripts were written in the narrow and careless style, the neighboring letters often overlap. Therefore the whole vertical group of characters can be interpreted as the single CC. To divide them, the text lines must be first isolated. If separate multiple lines intersect

the single CC, the latter should be divided into separate blocks.

The proposed text line segmentation methodology of handwritten documents faces the following challenges:

- each line of text can be at the arbitrary angle,
- the lines are not necessarily parallel,
- the distance between the lines of text converges to a minimum,
- accents may be above or below the line of text,
- the vast majority of neighboring CCs is connected, making their separation difficult.

The text lines detection is a widely explored domain, with multiple approaches already implemented (see Fig. 1). They will be briefly introduced below.

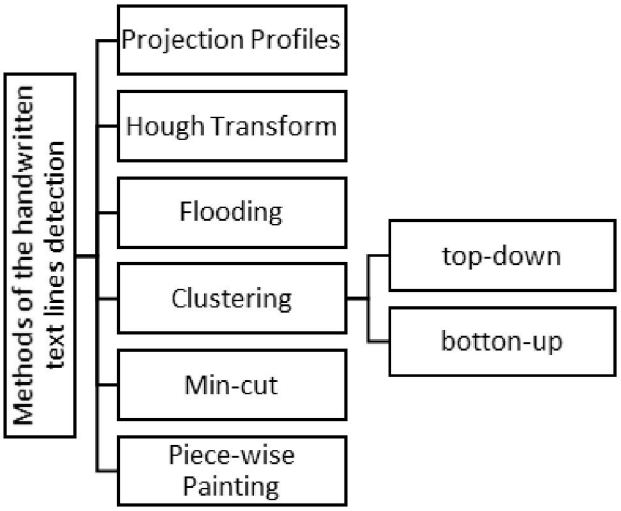

Fig. 1 – Taxonomy of the methods for the handwritten text lines detection

The projection profiles detect text lines based on the horizontal and vertical histograms of the input image. Their local extrema indicate the text line and the gap between the neighboring lines. Such approaches were initially susceptible to the angle of the written text. To increase their immunity to this effect, the input image is cut into horizontal and vertical stripes, for which calculated histograms have a higher accuracy than the single histogram calculated for the whole image [2, 3]. The disadvantage of these approaches is that pixels are treated cumulatively but context-free, i.e. only the number of ink pixels is obtained, without their position in the image. They correctly tackle the parallel, not overlapping handwriting and have small computational complexity. When overlapping of lines takes place, their accuracy decreases.

The Hough transform-based methods exploit the context processing. The input image is divided into labeled CC. Their set is used to construct the Hough transform accumulator, which is the matrix containing center points of CC, their minimal or maximal values (calculated along the vertical axis) [4, 5]. The local maxima in the accumulator are integers indicating the greatest number of curves intersected by the rotating lines, fixed at the accumulator point. Such values indicate the potential text lines. The improved Hough transform method divides the wider CC into smaller blocks and uses center points of each block to support the actual text lines [1]. Approaches from this group are characterized by the complex mathematical apparatus and high computational complexity. Recent achievements in this field lead to the high line detection accuracy even for the non-parallel handwriting. Unfortunately, the performance decreases with the increasing number of overlapping handwriting. The problem is suppressed by the introduced modification [6] described later in detail.

The flooding methods, based on the fuzzy adaptive Run-Length Encoding (RLE) are able to ignore single background pixels (i.e. the ones not covered by ink) [7, 8]. They are resilient to the askew handwriting. The idea of the algorithm is to trace the path of the ink in the long term, ignoring gaps between the letters. This way it is possible to detect the text line by focusing on the direction of the farther text pixels, not only on the closest ones. In [8] the image is represented in the form of one-dimensional column and row vectors, with ones and zeros indicating the presence or absence of the ink, respectively. If the gap between the letters (sequence of zeros) is small enough, they are flooded (exchanged with ones). The overlapping letters from neighboring lines are the problem here, as they may be incorrectly interpreted as the single text line.

The grouping methods are implemented in one of two ways: top down and bottom-up. The latter consist in connecting neighboring CCs into clusters and then creating the text line approximated by the curve instead of the straight line. This is effective if the text is bent. The advantage here is the correct processing of the askew handwriting. Unfortunately, the cluster may be constructed from a couple of neighboring lines, interpreted as the single line [10]. The top-down approach works conversely. First, the Foreground Pixel Density (FPD) is calculated. Next, the obtained result is processed by the isotropic LOG (Laplacean or Gaussian) filter, producing a set of images. They are then used to group elements identified in the images to assign them to the particular text lines. Clustering methods have the complex mathematical apparatus, require setting multiple parameters, adjusted for each image separately. Additionally, their computational complexity is high [11].

The cut text minimization methods consist in approximating the areas separating the neighboring

text lines by analyzing local minima of the histograms calculated for the binary input images. When locating the space between lines is not possible (because the letters overlap), the CC are cut and surrounded. Finding the local minimum in the histogram is not possible if the text is askew, which causes missing some lines [9].

Methods of intelligent painting are based on the analysis of CC. First, the input monochromatic image is processed by the median filter to eliminate small or unimportant elements for the future processing. Next, the filtered image is divided into stripes of the medium breadth of CC. Each pixel of the vertical image stripe is averaged with the mean value of the row in this stripe. The processed image is binarized using the Otsu algorithm. The morphological operation (dilation) is used to highlight areas where the actual text line should reside. The effectiveness of the method decreases with the increase of the overlapping text [12].

3. METHOD DESCRIPTION

The proposed method consists of three steps (presented in detail in the following subsections):

- preprocessing,
- Hough transform mapping,
- post-processing.

In the first step, the input image is binarized. As the result we get the two-colored image with the black points indicating text areas and white ones - empty spaces between them (background). This negated image is produced to facilitate arithmetical operations on pixels' colors. It is processed to extract CCs, SCCs and their features. In the second step, we use the Hough transform on both CC subsets (CCs and SCCs) to detect potential text lines. In the third step, we remove the false text lines. Afterwards, some CCs not being the part of any text line still remains. To them, we apply the procedure that checks if additional lines are present between the ones detected so far. The following subsections explain details of these operations.

3.1. PREPROCESSING

This stage consists of four steps. The first one is the binarization and extraction of CCs from the black-white image.

In [6] the Otsu binarization method was applied to transform the original image into the black and white version, further processed by the subsequent algorithms. It influences the accuracy of the text line detection, which is widely implemented on the scanned documents, where the parallel projection is used. In our approach the source of the text may be also the digital photocopy of the manuscript. Considering the non-uniform luminance of the document and distortion caused by the perspective of the data acquisition hardware, the more efficient method, based on the Gaussian filter [13], was proposed here. It is used to convert the input RGB image to the monochromatic, weighted version using the following transformation:

$$I_{gray} = \frac{1}{3} \cdot \left(I_{red} + I_{green} + I_{blue} \right) \tag{1}$$

Contrary to [13], here three binary images are generated, separately for each channel, eliminating the original hue and saturation retaining the luminance. The gray image is then processed by the Gaussian filter with the following parameters: $\sigma = 4.5$, $m_1 = 0.9$ and $m_2 = 0.1 \cdot 255 = 25.5$. They determine the range of the filter (the number of image pixels affected by the Gaussian function) and the intensity of the pixel modification regarding the center point. The resulting image I_G is compared against the monochromatic image, producing the binary image I_B with the following form:

$$I_{red_B} = \begin{cases} 1 & for \quad I_{gray} < m_1 \cdot I_G \\ 0 & otherwise \end{cases}, \tag{2}$$

Homogenous areas of the monochromatic image cause the noise, which can be eliminated by introducing the following condition:

$$\left| I_{gray} - I_G \right| > m_2 , \tag{3}$$

The fundamental components of the RGB image transformed, leading to "1" where the pixel fulfills (3) and "0" otherwise. Partial results (4)-(6) are next combined using the logical alternative (7). The only parameter of this transformation is the luminance resolution. For the scanned documents, the value of m_2 may increase from $0.05 \cdot 255 = 12.75$ (for the digital photography with the non-uniform luminance) up to $0.2 \cdot 255 = 51$. This allows for extracting more details without causing significant distortions.

$$I_{red_B} = \begin{cases} 1 & for \quad \left| I_{red} - I_G \right| > m_2 \\ 0 & otherwise \end{cases}, \tag{4}$$

$$I_{green_B} = \begin{cases} 1 & for \quad \left| I_{green} - I_G \right| > m_2 \\ 0 & otherwise \end{cases}, \tag{5}$$

$$I_{blue_B} = \begin{cases} 1 & for \quad \left| I_{blue} - I_G \right| > m_2 \\ 0 & otherwise \end{cases}, \tag{6}$$

$$I_B = I_{red_B} \vee I_{green_B} \vee I_{blue_B} , \tag{7}$$

The processed binarized image is next analyzed to find all connected components. Each CC is the rectangle surrounding the dense group of letters. The next steps require estimating the average width $\overline{w}$ and height $\overline{h}$ of the CC. The following requirement must be met to add the CC to the set S_1 (containing only characters, without other writing elements, such as accents):

$$cc_i \in S_1 \Leftrightarrow \left\{\left(\frac{1}{2}\cdot\overline{h} < h_i < 3\cdot\overline{h}\right) \wedge \left(w_i > \frac{3}{2}\cdot\overline{w}\right)\right\}, \quad (8)$$

where, h is the height of the i-th CC, and w – its width.

As this set is usually not enough to correctly detect all lines, the SCCs are additionally extracted from the image to increase the amount of input data for the Hough transform. To obtain them, we applied the histogram analysis for the image segments. The whole text is divided into vertical blocks, for which the histogram (presenting the number of white points in the subsequent areas of each block) is calculated (Fig. 2). The stripes of the histogram showing the locally maximal number of points indicate the y-coordinate of the SCC. The x-coordinate of the SCC is the middle position of the block.

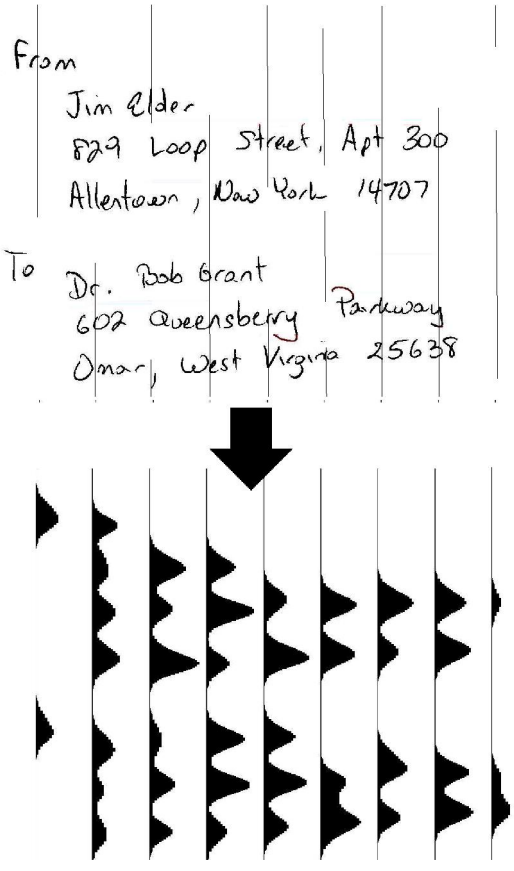

Fig. 2 – Illustration of the vertical histograms generation for the corresponding blocks of the binary image [1]

The efficiency of this step depends on the following factors:

- the breadth b of the vertical block dividing the binary image, calculated as $b = w \cdot n_3$. The real-valued n_3 should be as small as possible, because this way the block is resilient to the inclined text lines and the local histogram maximum is easier to detect. The best results are obtained for $n_3 = 2$.
- the minimal distance d between two local histogram maximal values required to regard them as the SCC y-coordinates. It is possible that between two actual text lines there is the line of emphasis marks, which could also form a local maximum. The proposed threshold $d = a \cdot n_4$ helps to avoid detecting the false lines. The best results were obtained for $n_4 = 1.5$.
- the number of histogram stripes, which is equal to the vertical image resolution (each stripe is calculated for one line of pixels – see Fig.).

3.2. HOUGH TRANSFORM MAPPING

In this step, a Hough transform is calculated from the CC and SCC subsets, as in the traditional approaches [4]. Because the transform is susceptible to the amount of the input data, each CC from S_1 longer than $\overline{w}$ is divided into smaller fragments with the length of $\overline{w}$. For each one, the x- and y-coordinate of the point being part of the text closest to the center of the fragment are calculated. Such points are used by the Hough transform (Fig.).

Fig. 3 – Partitioning of connected components into fragments

The transform is used to represent each center point in the original binary image by the curve in the polar coordinates, according to:

$$x \cdot \cos(\theta) + y \cdot \sin(\theta) = \rho, \quad (9)$$

The single curve is generated by the straight lines going through the corresponding CC center point in the original image. Because the infinite number of lines can go through such a point, only 360 of them are selected, differing by one degree in the angle related to the beginning of the Cartesian coordinates system. For each line, the perpendicular vector going through the center of the coordinates system is generated. Its length ρ and the angle θ between this line and the x-axis are recorded as the single point in

the polar coordinates system. This way for each center point from the original image, one curve is generated. The resulting diagram (the accumulator) is the array containing in each cell the number of curves' intersecting the CC for the particular ρ and θ. Because the Latin manuscripts are horizontally oriented, we consider only the points in the curve at the angles between 85 and 95 degrees.

Next, the areas where the maximum number of curves intersects are identified. They are considered as the line candidates. The areas where the intersections are searched for have the height of $0.2 \cdot \bar{h}$ [4] and one degree of width. The procedure for identifying the text lines is as follows:

1) We search for the cell $C_i(\rho,\theta)$ with the local maximal value in the accumulator (**Error! Reference source not found.**). Also, cells below and above the maximum, i.e. $(\rho-5,\theta)...(\rho+5,\theta)$ are considered [4], which is important in the next step.

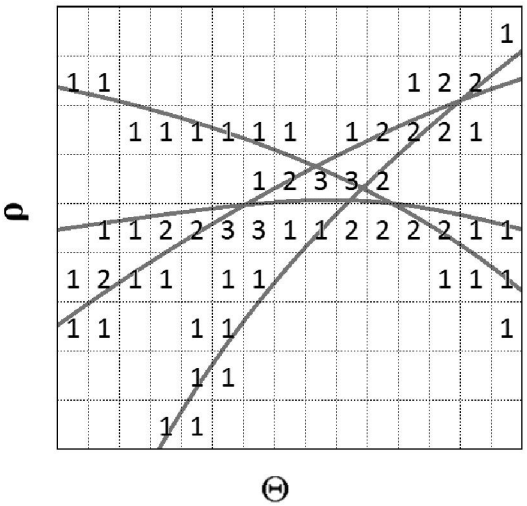

Fig. 4 – Example of the local maximum value in the accumulator array (Hough transform)

2) To determine whether the element of S_I belongs to the potential line (ρ,θ), half of the curves generated from its points must lie in the considered eleven cells. The value stored in each cell containing at least one curve belonging to the analyzed CC, is decreased by their number. For instance, if the cell with the value *5* contains two curves belonging to the considered CC, its value is decreased by 2.
3) The local maximum values allow for extracting two sets of results, depending on n_1 and n_2. If the number of intersections in the cell is greater than $n_1 = 8$ and the requirements from the second step are fulfilled, we can say with the full confidence that this is the text line. For such lines, their average angle $\bar{\theta}$ is calculated. Otherwise, if the number of intersections is greater than $n_2 = 4$ and the angle of the line candidate is different no more than *2* degrees from $\bar{\theta}$, this result is also considered a text line. If there are no more cells in the accumulator with the number of intersections greater than n_2, the procedure ends [1].

3.3. POST-PROCESSING

This stage consists of two operations. The first one is the elimination of overlapping potential text lines (as multiple detected lines may be constructed by mostly the same CCs, i.e. refer to the same text line). The second operation is the addition of the new line omitted by the Hough accumulator. To eliminate overlapping lines, the following method is implemented:

1) all intersections between potential lines are found in the input image.
2) vertical lines through the middle of the original image are created and the average distance between all intersections and this line are calculated.
3) If the distance between the intersection of two lines is less than the average, these candidates point at the same text line. One of them is eliminated, while the second is left intact.
4) the procedure is repeated until no more intersections are present.

Afterwards, the set of actual text lines is obtained. It is probable that some CCs are not associated with any line detected so far. Adding the lines not detected by the Hough transform is performed using the following method:

1) The average distance of all unconnected center CCs points from the nearest detected text line is calculated. The distance can be positive or negative, which allows for differing the points above and below the text line.
2) The CC can be used to create a new text line, if the distance of the half of its central points is greater than $\bar{h}$ from the nearest text line and below the average distance. To create a new line, a minimum of three points (potentially belonging to different CCs) is needed. Three central points from the separate bounding boxes ensure the correct line detection (see Fig.). Otherwise, it would be possible to create a false line from central points of the same CC.
3) The line $(y = ax + b)$ created in the second step is connected to all CC points constructing it. Because we assume the new text line is parallel to the nearest existing one, it takes its a value. We repeat this procedure until the set of central CC points is empty or does not contain enough points to create a line.

In Latin manuscripts (from the 15th century) the spacing between lines was very small. This was caused by the high price of parchment, therefore overlapping letters were very common. The size of the first CC subset for such a document would not be sufficient for the Hough transform. This may happen when the author does not write parallel text, so the α coefficient of the neighboring line can't be used. This problem is solved by SCCs we proposed. This way we collect more lines from the Hough domain.

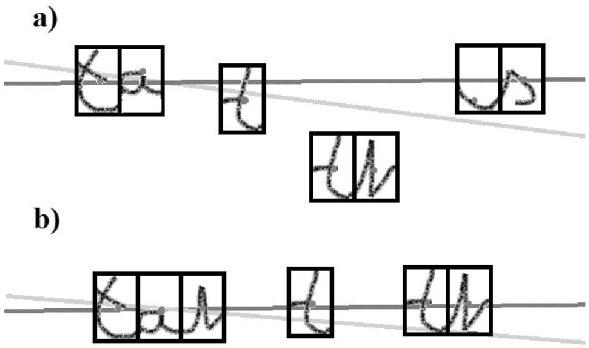

Fig. 5 – Illustration of the correct and incorrect line generation from at least three central points in CCs: the false line generated from two adjacent CCs (a) and from within central points of the single CC (b)

4. PERFORMANCE EVALUATION AND EXPERIMENTAL RESULTS

To evaluate the performance of a text line detection algorithm, we used visual criteria as in [14, 15]. This way the number of text lines in the manuscript is determined manually. The outcome the of computer algorithm is compared to the known actual number of lines. For this purpose we calculated the effectiveness in detecting actual text lines (True Positive – TP) and errors (False Negative – FN). First, the binarization was evaluated (Fig. 6). The presented outcomes refer to the modern document, although the text line was further performed on the Latin manuscripts. This is because the available historical documents were scanned using the professional device, ensuring the uniform luminance in the whole area of the image. Therefore no difference is visible in such examples, even the most primitive binarization method (such as Otsu) is suitable. In general, the less ideal acquisition conditions (such as the smartphone camera) may be available. In such cases the selected binarization strongly affects the initial image and should be optimized if possible. The proposed approach using the Gaussian filtering has significant advantages over the less sophisticated algorithms.

The subsequent text lines detection method was tested on selected ten pages of Latin Theologica Miscellanea documents dating from the fourteenth and fifteenth centuries [16]. It was collected from the Polish National digital library. The results of the experiment are shown in Table 1. Here "No." is the number of the document page, "$|L|$" is the number of actual lines, "TP_1" and "TN_1" (indicated in absolute numbers – No. and relative percentages - %) is the TP and TN ratio for the detection method described in [1], "TP_2" and "TN_2" is the ratio for the proposed approach, respectively.

Table 1. Results of lines detection

No.	$\|L\|$	TP_1		TP_2		TN_1	TN_2
		No.	%	No.	%		
26	37	30	81	34	92	2	0
24	28	14	50	27	96	4	0
22	31	23	74	30	97	5	1
21	32	25	78	30	94	4	1
14	48	24	50	44	92	6	1
12	48	26	54	35	73	9	6
11	52	26	50	44	85	9	2
10	47	25	53	29	62	6	1
9	42	37	88	42	100	3	4
5	59	41	69	54	92	5	1
∑	424	271	64	369	87	53	17
%	100	63.91509		87.0283		13	4

For tests the images were trimmed and their resolution reduced. We show that the proposed method detects text lines with an average of 87% efficiency. The methods used by other researchers achieve an average efficiency of 64% for the same pages. Our method is therefore more efficient by 23% and makes mistakes three times less than the method from [1]. We can see in Fig. and Fig. how supporting CCs affect the accuracy of the detected actual text lines (gray color) and are superior over the classical approach (black lines). In Fig. the simpler case is presented, where our algorithm shows superiority by detecting all the text lines. In Fig. the more difficult case can be seen, where some lines are not detected even by our approach.

The conducted experiments show that the Otsu algorithm is not the best choice for the image binarization. This is because the contrast is not identical in the whole document, the ink is also not distributed evenly along the paper. Therefore in the future research the adaptive approaches (adjusting to the local concentration of the ink) should be applied.

The optimal values of n_1, n_2, n_3 and n_4 coefficients were selected after [1]. As was proven there, such values ensure acceptable (if not optimal) performance of the SCC generation and lines detection, respectively. The future research should follow the systematic adjustment of their values.

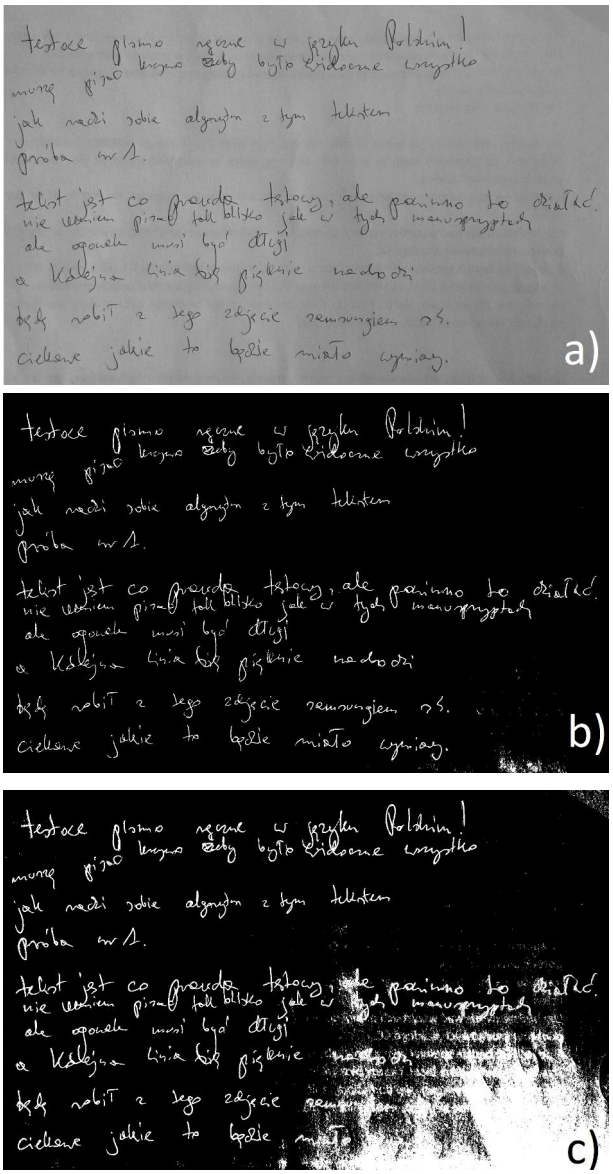

Fig. 6 – Comparison between the binarization methods: original document (a), results of Gaussian (b) and Otsu (c) binarization

5. CONCLUSIONS AND FUTURE WORK

The paper presented an improved method of detecting text lines in Medieval Latin manuscripts. Our contribution consisted in introducing the new binarization method and adding SCCs to the data used by the Hough transform, which results in the increased number of lines correctly detected and decreased number of errors compared to other approaches. Support CCs reduce the angle error detected in the actual line when the text lines are not parallel. The disadvantage of the method is its high demand on the processor power, as the processing of the single manuscript page takes about 10s. Therefore in the next step the efficiency optimization will be performed. The future work should involve the implementation of an advanced system for the manuscript author identification. The detection of text lines is fundamental for next stages.

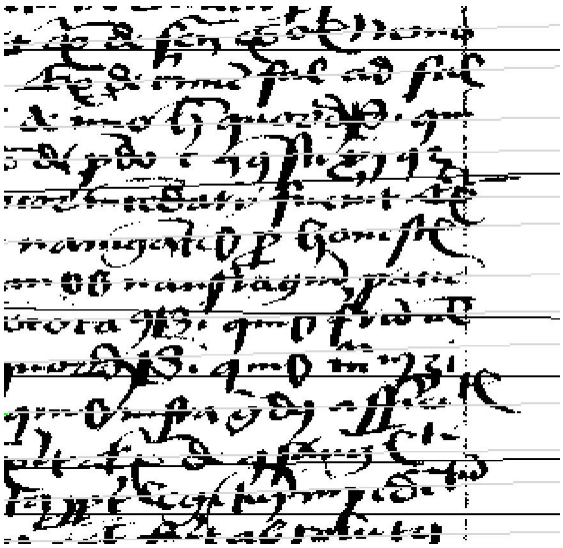

Fig. 7 – Detection of text lines in the simpler case. The classical method [1] – black lines; proposed method – gray lines

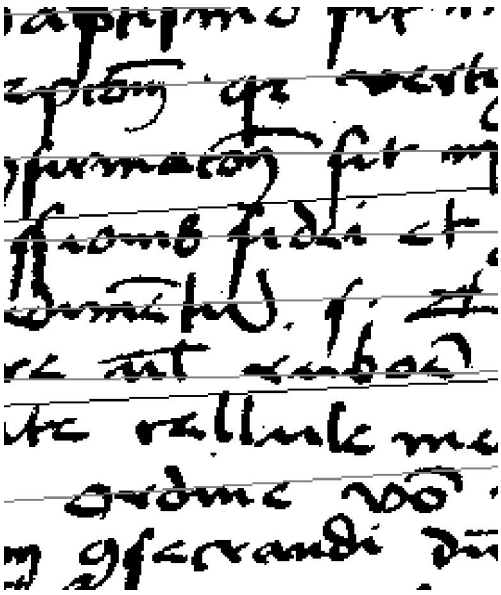

Fig. 8 – Detection of text lines the most difficult case. The classical method [1] - black lines; proposed method - gray lines.

REFERENCES

[1] G. Louloudis, B. Gatos, I. Pratikakis and K. Halatsis, "A block-based Hough transform mapping for text line detection in handwritten documents," in *Proceedings of the Tenth International Workshop on Frontiers in Handwriting Recognition,* 2006.

[2] 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),* 1999, pp. 749-752.

[3] M. Arivazhagan, H. Srinivasan and S. Srihari, "A statistical approach to line segmentation in handwritten documents," Vol. 6500, pp. 65000T, 2007.

[4] L. Likforman-Sulem, 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,* 1995, Vol. 2, pp. 774-777.

[5] Y. Pu and Z. Shi, "A natural learning algorithm based on Hough transform for text lines extraction in handwritten document," in *Proceedings of the 6th International Workshop on Frontiers in Handwriting Recognition*, 1988, pp. 637-646.

[6] J. L. Pach and P. Bilski, "A robust text line detection in complex handwritten documents," in *Proceedings of the 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2015),* Warsaw, Poland, 2015, pp. 271-275.

[7] B. Gatos, N. Stamatopoulos and G. Louloudis, "ICDAR2009 handwriting segmentation contest," *International Journal on Document Analysis and Recognition (IJDAR),* Vol. 14, pp. 25-33, 2011.

[8] F. M. Wahl, K. Y. Wong and R. G. Casey, "Block segmentation and text extraction in mixed text/image documents," *Computer Graphics and Image Processing,* Vol. 20, pp. 375-390, 1982.

[9] C. Weliwitage, A. Harvey and A. Jennings, "Handwritten document offline text line segmentation," in *Proceedings of the International Conference on Digital Image Computing: Techniques and Applications, (DICTA'05),* 2005, pp. 27-27.

[10] H. I. Koo and N. I. Cho, "Text-line extraction in handwritten Chinese documents based on an energy minimization framework," *IEEE Transactions on Image Processing,* Vol. 21, pp. 1169-1175, 2012.

[11] Y. Tang, X. Wu and W. Bu, "Text line segmentation based on matched filtering and top-down grouping for handwritten documents," in *Proceedings of the IAPR 11th International Workshop on Document Analysis Systems (DAS),* 2014, pp. 365-369.

[12] A. Alaei, P. Nagabhushan and U. Pal, "A new text-line alignment approach based on piece-wise painting algorithm for handwritten documents," in *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR'2011),* 2011, pp. 324-328.

[13] H. I. Koo and N. I. Cho, "State estimation in a document image and its application in text block identification and text line extraction," in Proceedings of the International Conference on *Computer Vision (ECCV'2010)*, Springer, 2010, pp. 421-434.

[14] Z. Shi, S. Setlur and V. Govindaraju, "Text extraction from gray scale historical document images using adaptive local connectivity map," in *Proceedings of the Eighth International Conference on Document Analysis and Recognition,* 2005, pp. 794-798.

[15] Z. Shi and V. Govindaraju, "Line separation for complex document images using fuzzy runlength," in *Proceedings of the First International Workshop on Document Image Analysis for Libraries,* 2004, pp. 306-312.

[16] Anonymous, "Miscellanea theologica," 2015.

Piotr Bilski PhD, DSc *was born in 1977 in Olsztyn, Poland. He graduated from Warsaw University of Technology, Institute of Radioelectronics, obtaining MSc degree in 2001 (with honors), PhD degree in 2006 (with honors) and DSc degree in 2014 in computer science. Currently he is an Associate Professor in the Institute of Radioelectronics and Multimedia Technologies, Warsaw University of Technology. His scientific interests include diagnostics of analog systems, design and analysis of virtual instrumentation, application of artificial intelligence and machine learning methods to the acoustics and environmental sciences. He is the member of IEEE, IMEKO TC10 and POLSPAR and reviewer for such journals like Measurement, IEEE Transactions on Instrumentation and Measurement, Expert Systems with Applications.*

Jakub Leszek Pach *was born in 1988 in Chorzow, Poland. He received in MSc degree in computer science in 2012 from Warsaw University of Life Sciences. Currently he pursues his PhD degree in Warsaw University of Technology, Institute of Radioelectronics and Multimedia Technologies. His research interests include artificial intelligence methods in image recognition, analysis of historical manuscripts and software engineering.*