Int. J. Appl. Math. Comput. Sci., 2017, Vol. 27, No. 1, 195–206
DOI: 10.1515/amcs-2017-0014

PROJECTION–BASED TEXT LINE SEGMENTATION WITH A VARIABLE THRESHOLD

ROMAN PTAK [a], BARTOSZ ŻYGADŁO [a], OLGIERD UNOLD [a,*]

[a]Department of Computer Engineering
Wrocław University of Science and Technology, Wyb. Wyspiańskiego 27, 50-370 Wrocław, Poland
e-mail: {roman.ptak,bartosz.zygadlo,olgierd.unold}@pwr.edu.pl

Document image segmentation into text lines is one of the stages in unconstrained handwritten document recognition. This paper presents a new algorithm for text line separation in handwriting. The developed algorithm is based on a method using the projection profile. It employs thresholding, but the threshold value is variable. This permits determination of low or overlapping peaks of the graph. The proposed technique is shown to improve the recognition rate relative to traditional methods. The algorithm is robust in text line detection with respect to different text line lengths.

Keywords: document image processing, handwritten text line segmentation, projection profile, off-line cursive script recognition.

1. Introduction

Document image segmentation is an important problem in document recognition. This concerns both machine writing and handwriting. However, other problems are encountered in both cases. Multi-column layouts and multi-line text are often used in printed documents. This kind of documents has multi-skew text and a combination of text and images. These problems are typically not present in handwritten documents, because they represent often a one-column document template. The main challenge in handwritten documents is different: variation of the text skew in each text line, while the most important one is touching and overlapping text-line elements between neighbouring lines. Furthermore, single words or short text lines may appear between the principal text lines. Although algorithms for printed document segmentation have been proposed (O'Gorman, 1993; Hull, 1998), their use in the processing of handwritten documents has been ineffective.

Handwriting processing and pattern recognition may consist of several stages: pre-processing, segmentation, feature extraction and analysis, classification and interpretation. Text line separation belongs to the second of these stages. This is not a trivial issue because handwriting can vary in shape, size, orientation, alignment, foreground and background colour, and texture. These variations make the process of word detection complex and difficult.

There are several line segmentation methods: projection-based, smearing, grouping, and methods based on the Hough transform, stochastic, etc.

Here we present a method to segment text lines based on horizontal projection profiles. The paper is structured as follows. Section 2 summarizes existing techniques, focusing on projection-based methods. Section 3 describes the proposed algorithm. The experimental evaluation of our method is presented in Section 4, in which it is compared with three other techniques, based on projection profiles. Section 5 draws conclusions and includes a proposal for further work.

2. Related work

Several methods of line segmentation have been developed. Likforman-Sulem *et al*. (2007) presented a survey of the methodologies proposed in the literature. Razak *et al*. (2008) made a comprehensive review of the methods of offline handwritten text line segmentation.

2.1. Methods used in segmentation. Generally, in segmentation, two strategies can be used: top-down

*Corresponding author

and bottom-up. Hybrid methods which combine both strategies, are also used.

Top-down approaches perform recursive XY-cuts, e.g., (i) horizontal and vertical projection profile analysis, (ii) white streams (spaces) analysis, (iii) the run-length smearing (smoothing) algorithm.

Methods based on projection profiles are discussed in the next sub-section. Analysis may refer to distributions of both foreground and background pixels.

For printed documents, smearing methods can be applied. An example of this type of algorithm is the run-length smoothing algorithm (RLSA) (Wong *et al*., 1982). The black pixels, representing foreground in the binary image of handwriting, are linked together along the horizontal direction if their distance is below a predefined threshold. The direction of smearing should be consistent with that of the line of handwriting. A variant of this method adopted to gray level images is described by LeBourgeois (1997). There are also modifications of the RLSA used for handwriting recognition (e.g., Sarkar *et al*., 2011).

The concept of the Hough transform is employed in the field of document analysis for many purposes such as skew and slant detection and text line segmentation (Likforman-Sulem *et al.*, 1995; Louloudis *et al.*, 2008; 2009; Alaei *et al.*, 2011). The Hough transform is a popular technique for finding straight elements in images. It can be used to determine the slope of elements. The pixel- and the block-based Hough transform can be distinguished (Louloudis *et al*., 2008). Hough transform-based methods can cope with documents with variations in the skew between lines (Likforman-Sulem *et al*., 1995; Pu and Shi, 2000).

In document image analysis, morphological filters have been also used for image segmentation. In the work of Papavassiliou *et al*. (2010) a method based on binary morphology was proposed. It uses morphological dilation and opening. The dilation is applied to determine text line components through joining close and horizontally overlapping regions. The generalized foreground rank openings prevent a merge in the vertical direction.

A novel "water flow" text line segmentation method was proposed by Basu *et al*. (2007). It assumes that hypothetical water flows from both sides of the image area. The stripes of areas left unwetted on the image are labelled for extraction of text lines. This algorithm was extended and further improved (see Brodić and Milivojević, 2011; Brodić, 2012; 2015).

Bottom-up methods start from low level visual objects, i.e., pixels, and iteratively group them into larger regions. There are several known approaches, for instance, connected component extraction and region grouping. The former scans an image and groups its pixels into components based on pixel connectivity features, e.g., pixel intensity values, colors. Grouping methods consist in building alignments by aggregating pixels, connected components or other units. Units are joined together and form alignments representing lines of text. In the work of Likforman-Sulem and Faure (1994) an iterative method based on perceptual grouping for forming alignments was developed.

Furthermore, the following methods may be mentioned: the repulsive-attractive network method (Öztop *et al*., 1999), the stochastic method (Tseng and Lee, 1999) and a graph-based method (an example is the docstrum method (O'Gorman, 1993)).

2.2. Projection-based methods. Projection profile is a one-dimension representation of a two-dimensional image; see Fig. 1. Projections can be horizontal. In such data representation—a horizontal projection profile diagram of an image (also presented as the histogram)—the amount of information is reduced. The values of the diagram represent the density distribution of handwriting. The method consists in calculating, for each horizontal line of pixels, the number of foreground pixels. Projection profiles may be applied to shape analysis (Pavlidis, 1982). This technique has been widely used in segmentation for machine printed documents (Ha *et al*., 1995). The most difficult problem in the area of document understanding and writing recognition is segmentation of cursive handwriting.

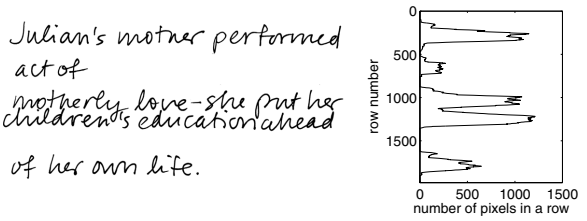


Fig. 1. Horizontal projection of a handwriting sample.

Antonacopoulos and Karatzas (2004) proposed a method based on horizontal projection applied to printed records which have regularly spaced text lines. In this case, there is a possibility to determine a parameter called the benchmark distance. First, horizontal projection is calculated. Each minimum of the obtained profile curve is a potential segmentation point. A histogram of the distances between subsequent separators is then constructed, and the most frequently encountered distance is identified as a benchmark one. Potential points are then scored, based on whether the distance to their neighboring separator points is consistent with the benchmark one. The highest scored segmentation point is admitted as the first true textline separator and the rest of the separators are appointed based on the benchmark distance. This method is suitable for machine printed texts.

A projection performed profile which results in text

line positions (dos Santos *et al*., 2009). To divide the lines into individual regions, a threshold is applied. This threshold is dynamically calculated and it is proportional to the average length of the lines in the document. The threshold value separates text lines. In the next step, a false line exclusion algorithm is applied. The lines with the height below a pre-determined threshold are removed. The latter threshold value is proportional to the average height of the text lines in the whole document. In this case, two heuristically determined parameters (thresholds) are used.

Manmatha and Srimal (1999) use a modified version of the projection algorithm extended to gray scale images. The input to the algorithm is a gray level document image. Line segmentation involves detecting the positions of the local maxima. The projection profile is smoothed with a low pass Gaussian filter to reduce sensitivity to noise. The local maxima are then obtained from the first derivative of the projection profile.

This line segmentation method is robust to variations in the size of the lines. There are, however, a few drawbacks. For instance, there are problems related to short text lines, which can give low peeks. Very narrow text lines may be omitted and overlapping lines may be indistinguishable. There are several other obstacles such as skew or moderate fluctuations of the text lines. A possible solution is to apply the partial projection method. The image may be divided into vertical strips and profiles computed inside each strip (Zahour *et al*., 2001).

Shapiro *et al*. (1993) used a projection at an angle according to the slope of the text lines. This angle is determined by the Hough transform. This eliminates the problem of handwriting slope.

Marti and Bunke (2001a; 2001b) used a modified method, where the numbers of transitions from background to foreground pixels are counted along horizontal lines through the character image. However, this does not change the fundamental image of the projection profile.

Methods based on projections are also used in other applications besides handwriting recognition, e.g., in medical diagnostics (Cierniak, 2014). In this case (computer tomography), we are dealing with the problem of image reconstruction from projections.

2.3. Filtering. Some of these projection-based methods use graph filtering. The raw graph is smoothed by a filter to eliminate outliers and noise. The profile curve can be smoothed by a Gaussian or median filter to eliminate local extrema. These elements (local maxima, local minima) may cause false detections in algorithms.

Figure 2 shows an example of averaged image projection. The image is smooth for easy classification. There are no unnecessary peaks there. However, unfortunately, also in this case there is no possibility of selecting one classification threshold. The dashed line shows the maximum possible threshold for detecting the second text line, which is not sufficient to distinguish between the third and fourth text lines.

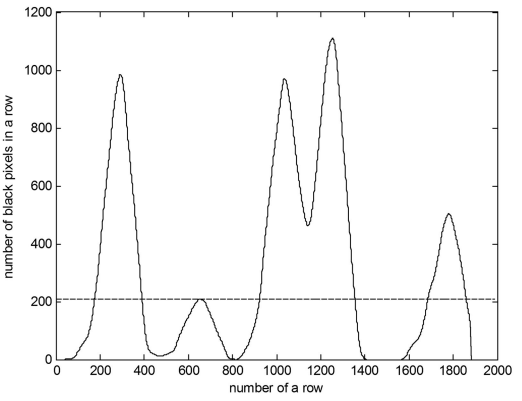


Fig. 2. Averaged projection of the image from Fig. 1.

In order to overcome that disadvantage, the algorithm presented in the next section is proposed.

3. Text line segmentation

The proposed method, as most text line segmentation algorithms, consists of several steps. The first one which must take place before the actual algorithm is executed is pre-processing.

3.1. Pre-processing. Pre-processing is the first stage of document analysis. Its purpose is to improve the quality of the processed image. A method significantly increasing the visibility of some hardly recognisable objects was used by Fabijańska *et al*. (2014). However, text on handwriting images is well visible and they only need noise cleaning. Thus, pre-processing used in the proposed method consists of the following steps:

1. conversion of a colour image to gray scale,
2. binarization,
3. noise reduction.

The basis of the proposed method, as well as other examined algorithms, is the projection profile of a grayscale or binary image. The grayscale image is determined by calculating a weighted sum of R, G and B components for every pixel of the colour image:

$$S = 0.2989 \times R + 0.5870 \times G + 0.1140 \times B.$$

The weights of the sum are derived from the Y'UV and Y'IQ models used by PAL and NTSC.

The second step of pre-processing is binarization. A grayscale image is converted to a binary one using thresholding. Pixels of luminance greater than the threshold are replaced with white and other pixels with black. The value of the threshold is calculated using Otsu's method (Otsu, 1975).

After binarization, reduction of noise is performed. Noises in the background are removed by employing the morphological opening operation. Gaps in the text area, caused by, for example, variable pen pressure while writing, are reduced using the closing operation.

The processed data can be subjected to an algorithm of segmentation.

3.2. Proposed method. In order to avoid the problems described in the previous section, we proposed a new algorithm based on the density distribution diagram. A global-to-local strategy was used in segmentation. This is achieved by analysis of the horizontal projection profile. The values of the projection profile are not normalized.

The main purpose of the new method is extraction of globally significant peaks of the graph. The algorithm is based on thresholding, but the threshold is not constant in all range of arguments. The variable threshold permits determination of low or overlapping peaks of the graph (Fig. 2). It is also resistant to small, insignificant local maxima.

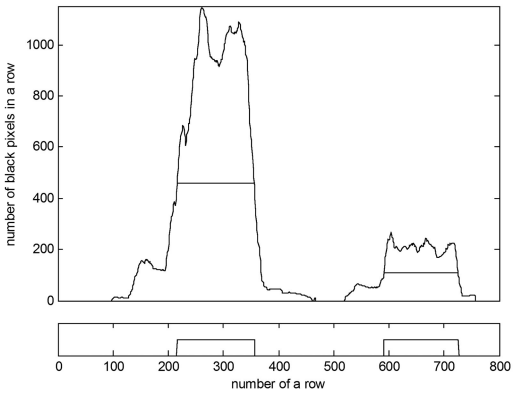


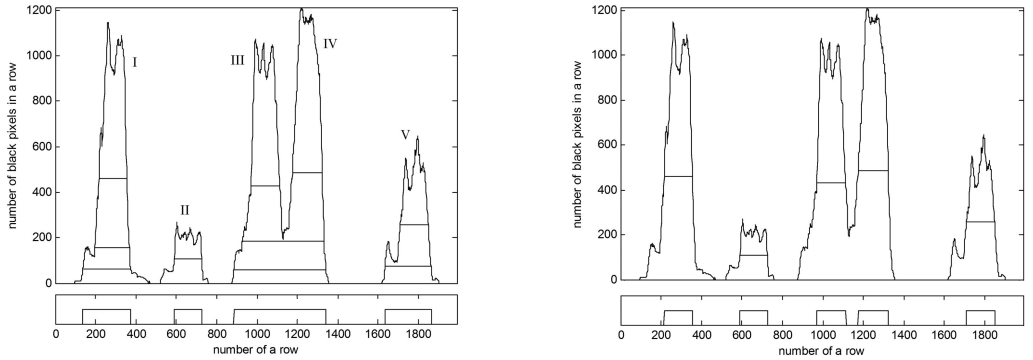
Fig. 3. Determination of widths of two first peaks of the projection profile from Fig.1. The threshold is different for each peak and is proportional to its height. Horizontal segments on the projection profile indicate intervals determined as peaks. The graph on the bottom presents set B, which is a sum of all found intervals.

The algorithm works as follows (see Algorithm 1). The input of the algorithm is a 2D text image I_{in} and two parameters: t, the relative threshold, and w, the window size of the filter. The output is the image I_{out} with text line separators marked on it. After pre-processing the projection profile is counted and filtered using the moving average filter. Filtering reduces local extrema to be analyzed. All points of the projection profile are sorted by their y-values. The points are processed in a descending sequence starting from the point which has the maximum y-value. For each point the width of the peak to which it belongs is determined at a certain height. Its value in proportion to the height of the peak is equal to the threshold t. The value of t is constant, but it is relative and thresholding is performed on a height t_α depending on the maximum y-value of the peak. Thus, the threshold is floating. The width of a peak is defined as the size of the range of arguments having the relative y-value greater than the threshold t (lines 7–9, Fig. 3). If a range R does not overlap any of the previously determined ranges (set A), it is accepted as a text line and added to set B (lines 10 and 11). Otherwise it is rejected. This prevents

Algorithm 1. Algorithm with a variable threshold.

Require: $I_{\text{in}}, I_{\text{out}}, H, X, n, A, B, R, S, \alpha, t, t_\alpha$ { I_{in}, input text image; I_{out}, output image with text lines separators; H, projection profile; $X = [1, n]$, domain of H; n, height of an image in pixels; A, set of all checked points; B, set of intervals corresponding to peaks of the projection profile. One interval denotes one peak or equivalently one text line (Fig. 3) (B^c is a complement of a set B); R, range of width of a peak; S, set of separators between text lines; α, parameter equal to 0.1; t, relative threshold within the interval (0, 1); t_α, absolute value of the threshold in a given iteration of the algorithm.}

1: Count projection profile H of I_{in} in the horizontal direction.
2: Sort X in descending order of $H(X)$ values.
3: $A \leftarrow \phi$, $B \leftarrow \phi$, $i \leftarrow 1$
4: **while** $H(X(i)) > \alpha \max(H)$ **do**
5: **if** $X(i) \notin A$ **then**
6: $t_a = tH(X(i))$
7: $R \leftarrow [x_1, x_2] : ((x_1 \le X(i) \le x_2)$
8: $\wedge(\forall x \in [x_1, x_2])(H(x) \ge t_a)$
9: $\wedge(H(x_1 - 1) < t_a) \wedge (H(x_2 + 1) < t_a))$
10: **if** $R \cap A = \phi$ **then**
11: $B \leftarrow B \cup R$
12: **end if**
13: $A \leftarrow A \cup R$
14: $i \leftarrow i + 1$
15: **end if**
16: **end while**
17: $j \leftarrow 1$
18: $I_{\text{out}} \leftarrow I_{\text{in}}$
19: **for each** Interval $[x_1, x_2]$ *in* B^c **do**
20: $S(j) \leftarrow x : H(x) = \min(H([x_1, x_2]))$
21: $I_{\text{out}}(S(j), *) \leftarrow [255, 0, 0]$
22: $j \leftarrow j + 1$
23: **end for**



(a) (b)

Fig. 4. Determination of ranges corresponding to text lines in the image from Fig. 1. In some cases overlapping ranges cause removal of spaces between adjacent peaks (3rd and 4th peak) and recognizing them as one text line (a); after rejection of overlapping ranges all text lines are determined correctly (b).

the connection of overlapping ranges, which would cause recognition of two or more text lines as one (Fig. 4).

The process terminates when the y-value of a given point is less than $\alpha = 0.1$ of the maximum value of the diagram (line 4). This value indicates the minimum relative height that a peak must have in order to be processed by the algorithm. Since a peak on a projection profile is equivalent to a text line, α should not be higher that the shortest text line on a page. Otherwise this text line would be omitted. On the other hand, it cannot be too low since there are redundant low peaks on the projection, which could be recognized as text lines (Fig. 5). Moreover, the higher it is, the quicker the algorithm terminates. Thus, α as high as possible, not exceeding the relative length of the shortest lines, is desired. The shortest lines on most samples do not exceed 20% of the width of the text. Only in one case was the relative length of the shortest line 12%. Thus, a slightly lower value of α equal to 10% was adopted. All found ranges in set B correspond to text lines and the minimum values of the regions between them are adopted as text lines separators (lines 19–23). The separators are marked in red on the output image (line 21).

Note that basic segmentation algorithms with a constant or variable threshold, e.g., depending on the average value of the projection profile, operate with linear time complexity. The addition of pre-processing increases computational time complexity. Blurring the projection profile diagram using a low-pass filter, e.g., using a Gaussian function, requires a convolution. The basic computational complexity of this type of algorithm is $\mathcal{O}(n^2)$. The fast convolution algorithms can reduce the cost of convolution to $\mathcal{O}(n \log n)$ complexity. In turn, the use of popular nonlinear filtering the median filter brings in its basic version the complexity of $\mathcal{O}(n^3)$ to $\mathcal{O}(n^2 \log n)$—depending on the sorting algorithm used,

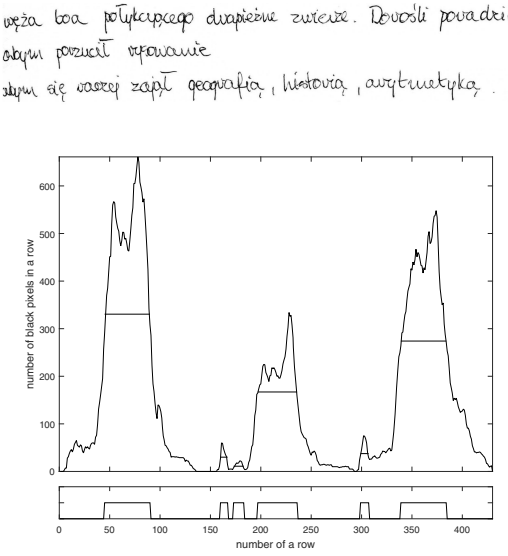


Fig. 5. Incorrect text line separation for $\alpha = 0$. The projection profile is rotated 90 degrees anti-clockwise. Although the image contains three text lines, the algorithm recognized six.

whereas fast median filter algorithms operate with the complexity $\mathcal{O}(n)$ (Perreault and Hébert, 2007).

Obtaining a horizontal projection profile curve has the complexity of $\mathcal{O}(nm)$, where the image size is $n \times m$. The proposed algorithm in the second stage (line 2) requires sorting with the complexity of $\mathcal{O}(n \log n)$ for quicksort. The main program loop (line 4) is executed n times. Determination of an interval $[x_1, x_2]$ (lines 7–9) in the i-th iteration of the loop requires k_i operations, which in the extreme case can be equal to n. However, this procedure is not executed if the given point has been checked before, i.e., if it belongs to set A (line 5). Thus,

the total number of operations in n iterations of the loop is equal to n: $\sum_{i=1}^{n} k_i = n$. In total, we get $\mathcal{O}(n \log n)$. The part of the algorithm determining the boundaries of line segments (lines 19–23) operates in linear time $\mathcal{O}(n)$. The entire algorithm (after projection count) has a computational complexity of the order $\mathcal{O}(n \log n)$.

4. Experimental results

We evaluated the performance of our algorithm on a database of unconstrained handwritten Polish documents and compared it with some projection-based algorithms: the Gaussian filter (Manmatha and Srimal, 1999), the median filter (Lim, 1990), and the Santos approach (dos Santos *et al*., 2009). In the following, we briefly describe the database and evaluation methodology, experiment design and statistical analysis, and the tuning of the algorithms' parameters, and then present the experimental results.

4.1. Research material. The research material consists of 60 dictated manuscripts divided into two sets with the total number of 1514 text lines. The first one contains ordinary samples of handwriting with a similar length of text lines in the document. The other set comprises more difficult images in terms of text lines segmentation. These include images with lines of different lengths which is common in texts including many paragraphs or dialogs. The text was written on paper without ruling lines. The images were scanned at a resolution of 300 dpi. Figure 6 shows samples of both kinds of examined images.

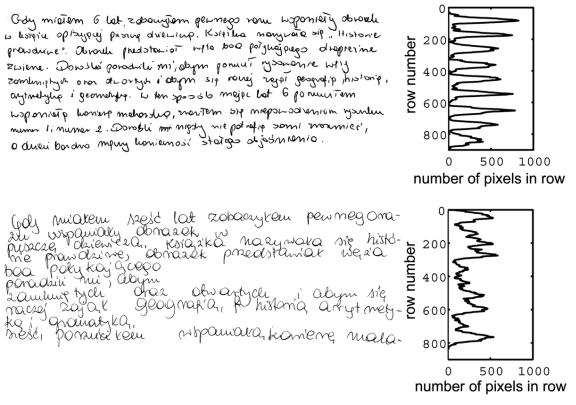


Fig. 6. Two examples of images with normal (top) and short (bottom) text lines and their projection profiles.

The average thickness of handwriting lines, calculated as the ratio of the number of pixels in the figure to the number of pixels in its skeleton, is 6.4 pixels. The research material and the Matlab source file are available freely online.[1]

4.2. Evaluation methodology. Several evaluation schemes may be used to assess the performance of text line segmentation algorithms. Many of the recent methods are based on a MatchScore table. A match score is a value between 0 and 1 that indicates the degree of conformity between the ON pixel sets of the result region R_i and the ground-truth region G_j. The score introduced by Yanikoglu and Vincent (1998) is defined as the percentage of the foreground pixels of G_j covered by R_i minus the percentage of the foreground pixels of R_i outside of G_j,

$$\text{MatchScore}(i,j) = \frac{T(R_i \cap G_j)}{T(G_j)} - \frac{T(R_i \setminus G_j)}{T(R_i)},$$

where $T(S)$ is a function that counts the number of foreground pixels of a set S. The MatchScore table is the basis of algorithm evaluation used in the ICDAR contest (ICDAR, 2013).

Although most of the recent evaluation methods are based on the MatchScore table, it is inappropriate for the proposed segmentation algorithm, as well as for the other ones compared in this paper. Firstly, the purpose of the examined methods is text lines localization without assigning the text fraction to the lines. Secondly, the straight-line separators produced by the projection profile methods cannot separate the overlapping and curved text lines without crossing them. Therefore, text line level evaluation appears to be more appropriate than pixel level one.

The performance of the algorithms was measured by determining the number of incorrectly identified text line separators. Two types of errors can be considered: missing and redundant separators (see Fig. 7).

The first one is equivalent to recognition of two text lines as one. Let s_i be the number of determined separators between the centers of the i-th and $(i+1)$-th text lines. The number of missing separators M_i is defined as

$$M_i = \begin{cases} 1 & \text{if } s_i < 1, \\ 0 & \text{if } s_i \geq 1. \end{cases}$$

The second type of error occurs when there are false separators which divide one text line into two or more. A separator is considered correct if it is located between the centers of two adjacent text lines and if it is the only one in this area. All extra separators are counted as false ones. Hence the number of redundant separators R_i is defined as

$$R_i = \begin{cases} 0 & \text{if } s_i \leq 1, \\ s_i - 1 & \text{if } s_i > 1. \end{cases}$$

[1] http://zio.iiar.pwr.wroc.pl/downloads/.

We add up all kinds of errors without distinguishing their type. Note that the MatchScore table also unifies the different types of errors to one measure.

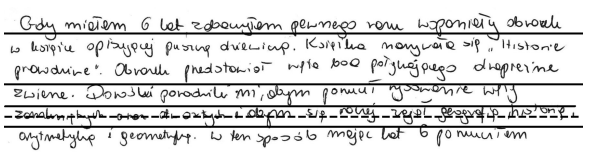


Fig. 7. Two types of text line segmentation errors. There is a missing separator between the second and the third line and a false separator in the fifth text line, marked with a dashed line.

4.3. Experiment design and statistical analysis. The proposed method was compared with three others based on the projection profile. Performance of the algorithms was measured using k-fold cross-validation (cv). In this method the original set of images is divided into k subsets and the validation is processed in k steps. In each step a single subset is used as testing data, whereas the sum of the remaining $k-1$ subsets is used as training data. The average of the obtained results in k steps of validation is the final evaluation of the algorithm.

To avoid high variance and non-zero bias of cv-based estimators (, 2010), a repeated cross-validation approach (Krstajic *et al.*, 2014) was used. In all experimental results, a *corrected resampled t-test* was applied (Nadeau and Bengio, 2003) instead of the standard t-test to test the difference in performance, at a 5% significance level. This corrects the dependencies in the estimates of the different data points, and is thus less prone to false-positives (type-I error) (Dietterich, 1998).

Nadeau and Bengio (2003) proposed the following statistic of the corrected resampled t-test:

$$t_c = \frac{\frac{1}{n}\sum_{j=1}^{n} x_j}{\sqrt{\left(\frac{1}{n}+\frac{n_2}{n_1}\right)\hat{\sigma}^2}},$$

where x_j is the difference of the performance quality between two compared algorithms on j-run ($1 \le j \le n$). We assume that in each run n_1 samples are used for training and n_2 samples for testing; $\hat{\sigma}^2$ stands for the variance of the n differences. This statistic obeys approximately Student's t distribution with $n-1$ degrees of freedom. According to Nadeau and Bengio (2003), as well as Bouckaert and Frank (2004), the corrected resampled t-test has the type-I error close to the significance level and (opposite to McNemar's test and the 5-times 2-fold cv test) low type-II error (i.e., the failure to reject a false null hypothesis). If we consider a test based on r-times k-fold cv, the statistic

$$t_c = \frac{\frac{1}{kr}\sum_{i=1}^{k}\sum_{j=1}^{r} x_{ij}}{\sqrt{\left(\frac{1}{kr}+\frac{n_2}{n_1}\right)\hat{\sigma}^2}}$$

has $k \times r - 1$ degrees of freedom and is called a *corrected repeated k-fold cv test*. To detect performance differentiation of the compared algorithms, we use a 10-times 5-fold cv scheme with 49 degrees of freedom. This scheme has been shown to have good replicability (Bouckaert and Frank, 2004). Note that to perform multiple comparisons involving a control method, we are supposed to control the family-wise error (FWER) (Demšar, 2006; Japkowicz and Shah, 2011; Trawiński *et al.*, 2012). The FWER is the probability of making a type-I error when testing many null hypotheses simultaneously. Several methods of relaxing the FWER have been proposed (Romano *et al.*, 2008). The Holm adjustment for the number of benchmarked learning schemes is applied (Holm, 1979).

4.4. Tuning parameters of the compared methods. In the training phase each compared algorithm is performed for each combination of parameters in a certain, reasonable range. Parameters giving the least amount of errors are adopted as optimal and used in the testing phase.

The proposed algorithm takes two parameters: the size of the median filter (w) and the threshold (t). Both of them influence the result of segmentation. However, there is a relation between them. For a given window size, the optimal value of the threshold depends on the value of the window size. There is a ridge in Fig. 8(a) showing a linear correlation between the two parameters. The algorithm achieved the best results for a threshold of about 0.9 of the maximal value of the projection profile for both sets of images. Optimal values of the window size were about 6 average thicknesses of the text (equivalent 42 pixels) for the first set of samples and 5 (32 pixels) for the second one. This indicates a visible difference, which is in line with the expectations. The difficulty of the images in the second set comes from short text lines, which results in a low height of peaks in the projection profile. Smoothing decreases this height even more, thus the level of smoothing should be lower in the case of the second set of images.

The Gaussian filter takes theoretically only the standard deviation (t_1) as a parameter. But in a digital implementation, it also requires another parameter, which is the size of the window (w_1). However, the research showed that it has no impact on the results in its effective range (Fig. 8(b)). The value of the chart is almost constant for a fixed sigma and the window size greater than a

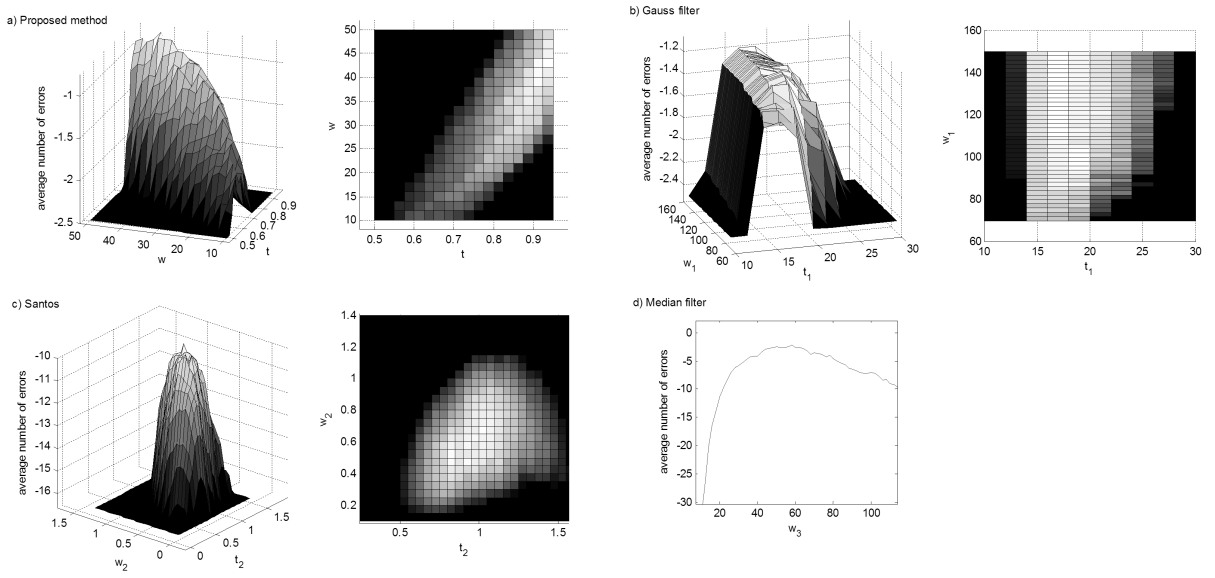


Fig. 8. Training phase of one step of k-fold cross validation on the first set of samples: proposed method (a), Gauss filter (b), Santos approach (c), median filter (d).

specific, initial value. The best results were obtained for a standard deviation of about 16 pixels. Variation in these values produces similar effects as in case of the median filter.

The first phase of the algorithm by dos Santos *et al.* (2009) is thresholding which requires one parameter: the threshold height (t_2) in proportion to the average y-value of the projection profile. The optimal results were obtained for the threshold of around 0.95 for the first set of images and 0.8 for the second one. Too low a threshold is not enough to distinguish some overlapping text lines, whereas too high a threshold causes omission of very short ones. The second parameter of the algorithm is the minimum width (w_2) that a peak must have in order not to be rejected. This value is calculated in proportion to the average width of all peaks of the projection profile. The optimal value for this parameter for subsets 1 and 2 was respectively 0.75 and 0.4 of the average width of peaks left after the thresholding phase. This shows a significant difference between optimal parameters for different kinds of research material.

The chart in Fig. 8(c) shows a strong dependency of the classification error on both parameters. There is a distinct global maximum indicating optimal combination of input parameters.

The median filter takes only one parameter: the size of the pattern of neighbors called the "window" (w_3). Figure 8(d) shows dependence between the window size and average number of errors. The optimal window size in 5 steps of validation varied between 44 and 58 pixels because of different training subsets in each step. Too wide windows do not distinguish some text lines because of missing significant local maxima. In contrast, a low degree of smoothing leaves too many local maxima, which are recognized as separate lines. The experiment indicated that lowering the window size by more than about 30 pixels causes a sharp increase in the number of errors whose limit is the total of local maxima in the projection profile.

4.5. Results and comparative analysis. Table 1 shows the means and standard deviations over 50 (10×5) cv fold results of the error rate for all compared algorithms on two datasets, i.e., a database with similar length of text lines (similar length database) and a database with mixed long and short text lines (mixed length database).

Table 1. Means and standard deviations over 10×5 error rate results on databases with a similar length of text lines and mixed long and short text lines.

Algorithm	Similar length database	Mixed length database
Proposed method	0.050 ± 0.029	0.081 ± 0.032
Santos	0.160 ± 0.043	0.228 ± 0.044
Gauss	0.057 ± 0.026	0.088 ± 0.030
Median	0.102 ± 0.045	0.116 ± 0.045

Figure 9 and Table 2, as well as Fig. 10 and Table 3, summarize the performances of the proposed method and compared approaches on images with a similar length of text lines and short text lines, respectively. The figures present boxplots representing the error rate obtained from 10×5 cv, whereas the tables give (unadjusted and adjusted

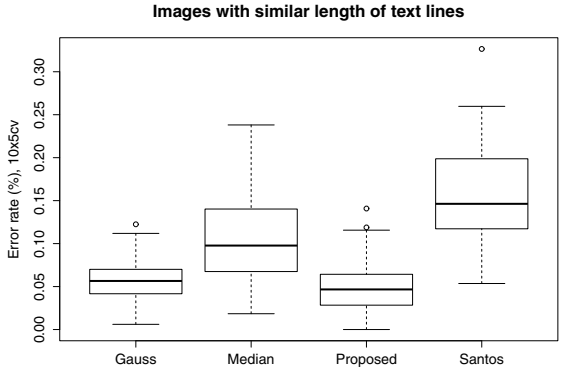


Fig. 9. Performance comparison of the proposed method, the Gaussian and median filters, and the Santos approach on images with a similar length of text lines. Box plots represent the error rate (%) obtained from 10×5 cross-validation.

Fig. 10. Performance comparison of the proposed method, the Gaussian and median filters, and the Santos approach on images with short text lines. Box plots represent the error rate (%) obtained from 10×5 cross-validation.

Table 2. p-values for the comparison of the proposed method (control algorithm) with the other methods on images with a similar length of text lines. The initial level of confidence $\alpha = 0.05$ is adjusted by the Holm procedure.

Proposed method vs.	Unadjusted p	Holm p
Santos	2.2565e-23	9.6867e-06
Gauss	1.9149e-03	3.7637e-01
Median	1.3271e-15	2.8154e-03

Table 3. p-values for the comparison of the proposed method (control algorithm) with the other methods on images with short text lines. The initial level of confidence $\alpha = 0.05$ is adjusted by the Holm procedure.

Proposed method vs.	Unadjusted p	Holm p
Santos	5.2482e-35	7.5528e-12
Gauss	1.4449e-03	3.6262e-01
Median	4.2883e-13	1.0533e-02

by the Holm procedure) p-values for the comparison of the proposed method (as the control method) with the remaining algorithms. Note that adjusted p for the Santos approach and the median filter over each database is lower than desired level of a confidence α, 0.05. These p-values indicate that there are significant performance differences between the proposed method and these two algorithms, hence confirming the superiority of the method with a variable threshold.

All methods work worst on the second, more difficult set of images. A higher error rate is mainly derived from the second type of error—missing text lines. Short lines are harder to detect, so that they increase the error of missing lines.

The statistical results show there is no difference between the proposed method and the Gaussian filter, both over handwriting texts with similar lines in terms of length and with mixed short and long lines. Both methods perform worse on images with mixed text lines, incorrectly identifying 12.16 ± 4.64 text lines (each testing fold contains about 150 text lines) in each testing folder—proposed method, and $13.28 \pm 4,36$ lines—Gaussian filter. For the same number of overall lines in each testing folder, the introduced method separates falsely 7.48 ± 4.26 text lines, and the Gaussian filter 8.6 ± 3.81 lines on handwriting texts with a similar length of text lines.

Note that the proposed algorithm does not require the use of the pre-processing of the density diagram (blurring), as it is a method by Manmatha and Srimal (1999) applying the Gaussian filter. The use of a Gaussian filter has the effect of strong smoothing so that it may lead to removal of peaks of the graph carrying information about a text line, which may be important especially in the case of short text lines.

The use of a linear filter with a Gaussian function may result in loss of information needed for proper segmentation of the lines of handwriting. On the left-hand side of Fig. 11 is shown the projection profile diagram of an image before filtering (top) and after applying Gaussian filtering (bottom). This figure shows local minima defining the segmentation points. The image on the right-hand side of Fig. 11 shows the result of incorrect image segmentation.

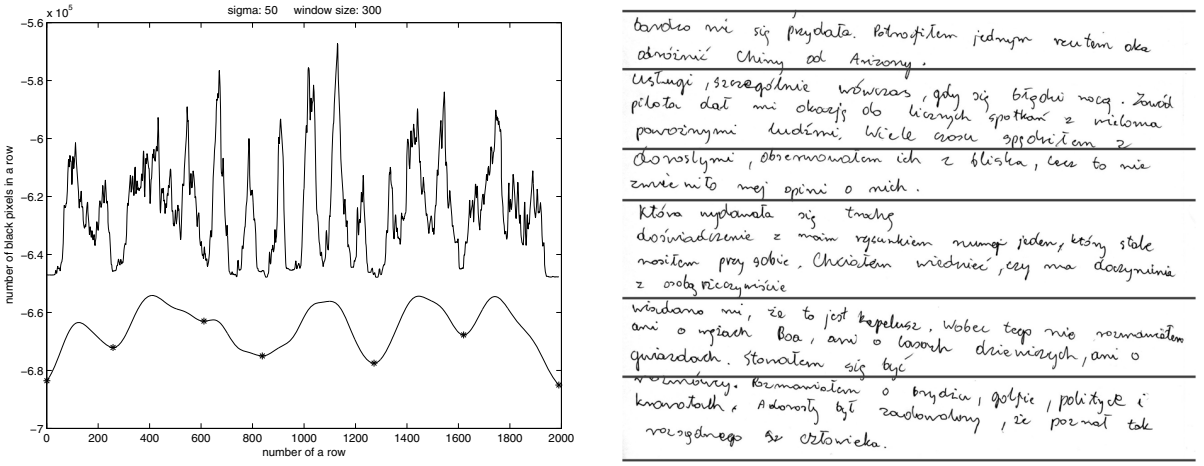


Fig. 11. Impact of applying a filter with a Gaussian function (standard deviation equal to 50 pixels) to the densit diagram and the line segmentation result.

5. Conclusion

The paper presented a text line segmentation algorithm based on the projection profile with a variable threshold. The threshold of the method is different for each peak and is proportional to its height. Experiments were made on a text database that consists of real text samples, both with a similar length of text lines, and harder to detect mixed in length text lines. Results of the experiments and a comparative analysis with other methods based on projection profiles: the Santos method, as well as the median and Gaussian filters, showed a decided advantage of the proposed approach over the first two methods in the area of text line segmentation. From a rigorous statistical point of view, the text line segmentation of the Gaussian filter is comparable with our proposal.

However, there are some drawbacks in using any kind of filter of text line segmentation. Note that state-of-the-art segmentation algorithms use a fixed threshold value. The disadvantage of simple thresholding is detection of small local maxima of the projection profile, which results in incorrect assignment of text lines. A solution to this problem may be reduction of noise with the use of a filter. Nevertheless, even after smoothing the graph of a projection profile, fixed thresholds do not work properly with short (significantly different from the average length) and overlapping text lines. Thus, some filtering methods search for local maxima instead of using thresholding. This approach requires a high degree of smoothing in order to eliminate all insignificant local maxima, which can also eliminate the correct ones. The proposed algorithm copes with both drawbacks. Its worth mentioning that in most segmentation algorithms data filtering is used.

The proposed algorithm also works well for small values of input averaging. Application of a variable threshold allows detection of significant peaks without using a strong filter. The proposed method can be developed to be applied for sloping text lines. The advantages of the method, dealing with overlapping and short lines, can be combined with a slope determination algorithm, e.g., the Hough transform. For curve text lines with a variable slope along the horizontal direction, a multi-part projection profile appears to be a good improvement of the algorithm. As a result of the application of the method, short local line separators would be obtained on narrow parts of the document, which, connected, may give fit global curve separators.

The limitation of the proposed method comes from drawbacks of projection profiling. Given information from a profile, which is calculated only in the horizontal direction, the algorithm does not deal well with slanting and curved text lines.

Acknowledgment

This research was supported by a statutory grant of the Wrocław University of Science and Technology.

R. Ptak, B. Żygało, and O. Unold designed the methodology and experiments, B. Żygało conceived and performed the experiments, O. Unold designed and performed the statistical data analysis, R. Ptak calculated the computational complexity, R. Ptak and O. Unold designed and supervised the study. All authors wrote and approved the final manuscript.

References

Alaei, A., Nagabhushan, P. and Pal, U. (2011). Piece-wise painting technique for line segmentation of unconstrained handwritten text: A specific study with Persian text documents, *Pattern Analysis and Applications* **14**(4): 381–394.

Antonacopoulos, A. and Karatzas, D. (2004). Document image analysis for World War II personal records, *International Workshop on Document Image Analysis for Libraries, 2004, Palo Alto, CA, USA*, pp. 336–341.

Arlot, S. and Celisse, A. (2010). A survey of cross-validation procedures for model selection, *Statistics Surveys* **4**: 40–79.

Basu, S., Chaudhuri, C., Kundu, M., Nasipuri, M. and Basu, D.K. (2007). Text line extraction from multi-skewed handwritten documents, *Pattern Recognition* **40**(6): 1825–1839.

Bouckaert, R.R. and Frank, E. (2004). *Evaluating the Replicability of Significance Tests for Comparing Learning Algorithms*, Springer, Berlin/Heidelberg, pp. 3–12.

Brodić, D. (2012). Extended approach to water flow algorithm for text line segmentation, *Journal of Computer Science and Technology* **27**(1): 187–194.

Brodić, D. (2015). Text line segmentation with water flow algorithm based on power function, *Journal of Electrical Engineering* **66**(3): 132–141.

Brodić, D. and Milivojević, Z. (2011). A new approach to water flow algorithm for text line segmentation, *Journal of Universal Computer Science* **17**(1): 30–47.

Cierniak, R. (2014). An analytical iterative statistical algorithm for image reconstruction from projections, *International Journal of Applied Mathematics and Computer Science* **24**(1): 7–17, DOI: 10.2478/amcs-2014-0001.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets, *The Journal of Machine Learning Research* **7**: 1–30.

Dietterich, T.G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms, *Neural Computation* **10**(7): 1895–1923.

dos Santos, R.P., Clemente, G.S., Ren, T.I. and Cavalcanti, G. D. (2009). Text line segmentation based on morphology and histogram projection, *10th International Conference on Document Analysis and Recognition, ICDAR'09, Barcelona, Spain*, pp. 651–655.

Fabijańska, A., Węgliński, T., Zakrzewski, K. and Nowosławska, E. (2014). Assessment of hydrocephalus in children based on digital image processing and analysis, *International Journal of Applied Mathematics and Computer Science* **24**(2): 299–312, DOI: 10.2478/amcs-2014-0022.

Ha, J., Haralick, R.M. and Phillips, I.T. (1995). Document page decomposition by the bounding-box project, *3rd International Conference on Document Analysis and Recognition, Montreal, Canada*, Vol. 2, pp. 1119–1122.

Holm, S. (1979). A simple sequentially rejective multiple test procedure, *Scandinavian Journal of Statistics* **6**(2): 65–70.

Hull, J.J. (1998). Document image skew detection: Survey and annotated bibliography, *Series in Machine Perception and Artificial Intelligence* **29**: 40–66.

ICDAR (2013). Handwriting Segmentation Contest, `http://users.iit.demokritos.gr/~nstam/ICDAR2013HandSegmCont/index.html`.

Japkowicz, N. and Shah, M. (2011). *Evaluating Learning Algorithms: A Classification Perspective*, Cambridge University Press, Cambridge, NY.

Krstajic, D., Buturovic, L., Leahy, D. and Thomas, S. (2014). Cross-validation pitfalls when selecting and assessing regression and classification models, *Journal of Cheminformatics* **6**(1): 10.

LeBourgeois, F. (1997). Robust multifont OCR system from gray level images, *4th International Conference on Document Analysis and Recognition, 1997, Ulm, Germany*, Vol. 1, pp. 1–5.

Likforman-Sulem, L. and Faure, C. (1994). Extracting text lines in handwritten documents by perceptual grouping, *in* C. Faure *et al.* (Eds.), *Advances in Handwriting and Drawing: A Multidisciplinary Approach*, Europia, Paris, pp. 21–38.

Likforman-Sulem, L., Hanimyan, A. and Faure, C. (1995). A Hough based algorithm for extracting text lines in handwritten documents, *3rd International Conference on Document Analysis and Recognition, Montreal, Canada*, Vol. 2, pp. 774–777.

Likforman-Sulem, L., Zahour, A. and Taconet, B. (2007). Text line segmentation of historical documents: A survey, *International Journal of Document Analysis and Recognition* **9**(2–4): 123–138.

Lim, J.S. (1990). *Two-Dimensional Signal and Image Processing*, Prentice Hall, Englewood Cliffs, NJ.

Louloudis, G., Gatos, B., Pratikakis, I. and Halatsis, C. (2008). Text line detection in handwritten documents, *Pattern Recognition* **41**(12): 3758–3772.

Louloudis, G., Gatos, B., Pratikakis, I. and Halatsis, C. (2009). Text line and word segmentation of handwritten documents, *Pattern Recognition* **42**(12): 3169–3183.

Manmatha, R. and Srimal, N. (1999). Scale space technique for word segmentation in handwritten documents, *in* M. Nielsen *et al.* (Eds.), *Scale-Space Theories in Computer Vision*, Springer, Berlin/Heidelberg, pp. 22–33.

Marti, U.-V. and Bunke, H. (2001a). On the influence of vocabulary size and language models in unconstrained handwritten text recognition, *6th International Conference on Document Analysis and Recognition, Seattle, WA, USA*, pp. 260–265.

Marti, U.-V. and Bunke, H. (2001b). Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system, *International Journal of Pattern Recognition and Artificial Intelligence* **15**(01): 65–90.

Nadeau, C. and Bengio, Y. (2003). Inference for the generalization error, *Machine Learning* **52**(3): 239–281.

O'Gorman, L. (1993). The document spectrum for page layout analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **15**(11): 1162–1173.

Otsu, N. (1975). A threshold selection method from gray-level histograms, *Automatica* **11**(285-296): 23–27.

Öztop, E., Mülayim, A.Y., Atalay, V. and Yarman-Vural, F. (1999). Repulsive attractive network for baseline extraction on document images, *Signal Processing* **75**(1): 1–10.

Papavassiliou, V., Katsouros, V. and Carayannis, G. (2010). A morphological approach for text-line segmentation in handwritten documents, *International Conference on Frontiers in Handwriting Recognition (ICFHR), Kolkata, India*, pp. 19–24.

Pavlidis, T. (1982). *Algorithms for Graphics and Image Processing*, Computer Science Press, Berlin/Heidelberg.

Perreault, S. and Hébert, P. (2007). Median filtering in constant time, *IEEE Transactions on Image Processing* **16**(9): 2389–2394.

Pu, Y. and Shi, Z. (2000). A natural learning algorithm based on Hough transform for text lines extraction in handwritten documents, *Series in Machine Perception and Artificial Intelligence* **34**: 141–152.

Razak, Z., Zulkiflee, K., Idris, M.Y.I., Tamil, E.M., Noor, M.N.M., Salleh, R., Yaakob, M., Yusof, Z.M. and Yaacob, M. (2008). Off-line handwriting text line segmentation: A review, *International Journal of Computer Science and Network Security* **8**(7): 12–20.

Romano, J.P., Shaikh, A.M. and Wolf, M. (2008). Control of the false discovery rate under dependence using the bootstrap and subsampling, *Test* **17**(3): 417–442.

Sarkar, R., Malakar, S., Das, N., Basu, S., Kundu, M. and Nasipuri, M. (2011). Word extraction and character segmentation from text lines of unconstrained handwritten Bangla document images, *Journal of Intelligent Systems* **20**(3): 227–260.

Shapiro, V., Gluhchev, G. and Sgurev, V. (1993). Handwritten document image segmentation and analysis, *Pattern Recognition Letters* **14**(1): 71–78.

Trawiński, B., Smętek, M., Telec, Z. and Lasota, T. (2012). Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms, *International Journal of Applied Mathematics and Computer Science* **22**(4): 867–881, DOI: 10.2478/v10006-012-0064-z.

Tseng, Y.-H. and Lee, H.-J. (1999). Recognition-based handwritten Chinese character segmentation using a probabilistic Viterbi algorithm, *Pattern Recognition Letters* **20**(8): 791–806.

Wong, K.Y., Casey, R.G. and Wahl, F.M. (1982). Document analysis system, *IBM Journal of Research and Development* **26**(6): 647–656.

Yanikoglu, B.A. and Vincent, L. (1998). Pink panther: A complete environment for ground-truthing and benchmarking document page segmentation, *Pattern Recognition* **31**(9): 1191–1204.

Zahour, A., Taconet, B., Mercy, P. and Ramdane, S. (2001). Arabic hand-written text-line extraction, *6th International Conference on Document Analysis and Recognition, Seattle, WA, USA*, pp. 281–285.

Roman Ptak received his MSc and PhD degrees in computer science from the Wrocław University of Science and Technology in 1998 and 2006, respectively, and the MA degree in history at the University of Wrocław in 2001. He is an assistant professor in the Department of Computer Engineering, Wrocław University of Science and Technology. His current research focuses on image recognition and computational intelligence and their application, as well as on spatio-temporal databases and data warehouses.

Bartosz Żygadło obtained his MSc in computer science from the Wrocław University of Science and Technology in 2012, where he then started his PhD degree. He spent part of his studies at Odessa National University, Ukraine. His field of interest includes image segmentation and 3D graphics.

Olgierd Unold is an associate professor in the Department of Computer Engineering of the Wrocław University of Science and Technology. He received an MSc degree in automation systems in 1989, an MSc degree in information science in 1991, and PhD and DSc degrees in computer science in 1994 and 2011, respectively. His research interests focus on adaptive machine learning methods.

Received: 17 January 2016
Revised: 16 August 2016
Re-revised: 17 October 2016
Accepted: 24 October 2016