

# TEXT LINE SEGMENTATION WITH WATER FLOW ALGORITHM BASED ON POWER FUNCTION

Darko Brodić \*

This manuscript proposes an extension to the water flow algorithm for text line segmentation. Basic algorithm assumes hypothetical water flows under few specified angles of the document image frame from left to right and vice versa. As a result, unwetted image regions that incorporate text are extracted. These regions are of the major importance for text line segmentation. The extension of the basic algorithm means modification of water flow function that creates the unwetted region. Hence, the linear water flow function used in the basic algorithm is changed with its power function counterpart. Extended method was tested, examined and evaluated under different text samples. Results are encouraging due to improving text line segmentation which is a key process stage.

**Key words:** optical character recognition, image processing, text line segmentation, water flow algorithm

## 1 INTRODUCTION

Text line segmentation represents a key step in off-line handwriting recognition in many optical character recognition systems [1]. It represents a labeling process which consists in assigning the same label to spatially aligned units, such as pixels, connected components or characteristic points [2]. Based on the obtained labels, text is divided into different regions. Each of these regions represents a text line. After finishing the text line segmentation, further document text image processing is introduced. Nevertheless, it provides the essential information for the consecutive steps such as skew correction, zone segmentation, and character recognition [3]. Although text line segmentation for machine or hand printed documents is usually seen as a solved problem [4], freestyle handwritten text line segmentation still remains an open research field [1–3]. Hence, it is a leading challenge in document analysis [2–3].

Several techniques for text line segmentation have been developed. They can be classified as follows [2]: projection based methods, Hough transforms methods, smearing methods, grouping methods, methods for processing overlapping and touching components, stochastic methods, and other methods.

Projection based methods have been mainly used for printed document text segmentation. However, they can be adapted for handwritten documents. These methods primarily use the horizontal projection profile, which is obtained by summing pixel values along the horizontal axis. It is achieved by finding its maximum and minimum [5–6]. Each local maximum represents the text line, while local minimum means interline spacing. However, multi-skew text lines contribute to higher complexity in finding the maximum and minimum values, while short

lines provide low peaks [2, 5]. Hence, the method failed to be efficient for multi-skewed text lines without radical modification and adaptation.

The Hough transform [7] is a widespread technique, among the rest, for finding straight lines in the images. Consequently, the image is transformed in the Hough domain. Possible alignments are hypothesized in Hough domain and validated in the image domain. The direction of the maximum variation is determined by a cost function. The “voting” function in Hough domain determines slope of the straight line [8]. Therefore, it needs adaptation to be used for the free-style handwritten text. However, the method is complex and computer time consuming.

Smearing methods have been developed for some time. Nevertheless, they become very efficient in the last decade. In smearing methods the consecutive black pixels along the horizontal direction are smeared [1, 2, 9–11]. This way, enlarged area of black pixels is formed. It is the so-called boundary growing area. Therefore, the white space between the black pixels is filled with black pixels. It is valid only if their distance is within a predefined threshold limit. Different methods that belong to this group has been proposed [3, 12–14]. They are primarily characterized as computer time and text segmentation efficient methods.

Grouping methods are based on building alignments by combining them into entities [15]. The units may be pixels or connected components [16], blocks or other characteristics such as salient points. These units are joined together to form alignments. The joining scheme is based on both local and global criteria used for checking consistency. Nevertheless, they need specific adaptation for freestyle handwritten text [16]. Furthermore, if the nearest neighbor element belongs to another line, then the

\* University of Belgrade, Technical Faculty Bor, Vojske Jugoslavije 12, 19210 Bor, Serbia, dbrodic@tf.bor.ac.rs

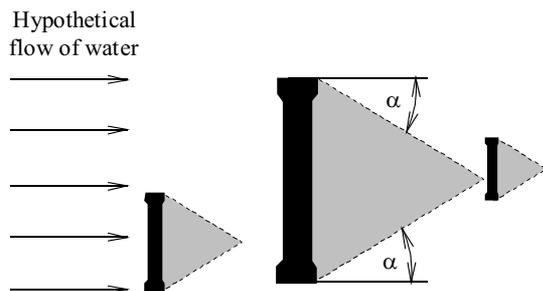


Fig. 1. Unwetted region definition (grey areas)

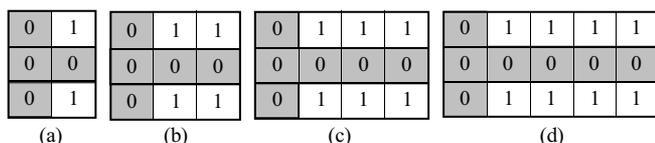


Fig. 2. Filter mask defined by flow angle  $\alpha$ : (a) —  $45^\circ$ , (b) —  $26.6^\circ$ , (c) —  $18^\circ$ , and (d) —  $14^\circ$

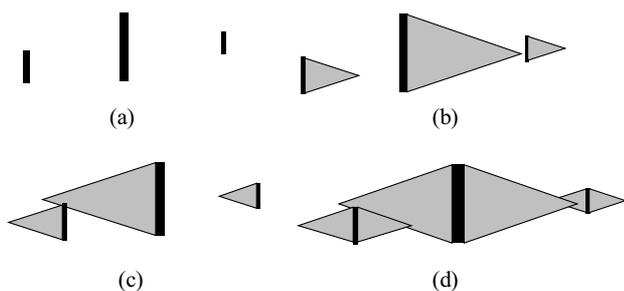


Fig. 3. Water flow algorithm applied to sample text: (a) — initial text consists of three letters I, (b) — water flows from left to right, (c) — water flows from right to left, and (d) — union of the water flows

nearest neighbor joining scheme will fail to group complex handwritten units.

Method for overlapping and touching components detects such components during the grouping process. At present, the conflict occurs between the two alignments [17]. Further, it applies a set of rules to label overlapping or touching components. The rules use as features the density of dark pixels of the component in each alignment region, alignment proximity and positions of both alignments around the component. The frontier segment position is determined by analyzing the component horizontal projection profiles. If it includes two peaks, the cut will be done in the middle of them. Otherwise, the component will be cut into two equal pieces. These methods are characterized as complex, demanding and computer time consuming.

Stochastic method is based on probabilistic algorithm, which accomplished non-linear paths between overlapping text lines. These lines are extracted through hidden Markov modeling (HMM) [18]. This way, the image is divided into small cells. Each one of them corresponds to

the state of the HMM. The best segmentation paths are searched from left to right. In the case of touching components, the path of highest probability will cross the touching component at points with as less black pixels as possible. The method may fail in the case that contact point contains a lot of black pixels. Furthermore, these methods have been classified as complex ones.

Graph-based method belongs to the other methods. It is based on a shortest spanning tree search [19]. The principle of the method consists of building a graph of main strokes of the document image and searching for the shortest spanning tree of this graph. However, it presumes that the distance between the words is less than the distance between two adjacent text lines. Hence, it requires an adaptation. Furthermore, graph cuts based framework that exploits a swap algorithm to segment document images is given in [20]. Firstly, text block is segmented into lines using the projection profile approach. Consequently, the framework enables learning of the spatial distribution of the components of a specific script. Moreover, they can use both corrections made by the user and any segmentation quality metric to improve the segmentation quality.

Water flow algorithm [21] is classified as smearing methods. It assumes hypothetical water flows under a few specified angles of the document image frame from left to right and vice versa. As a result of the basic water flow algorithm, unwetted image frames are established. They are key elements in the text line segmentation process. The basic algorithm is improved in [22]. In this paper, the algorithm is further extended and implemented.

## 2 WATER FLOW ALGORITHM

Document text image is obtained as product of original image scanning. It is digital image represented by matrix  $\mathbf{D}$  with  $M$  rows,  $N$  columns, and intensity with  $L$  discrete levels of grey.  $L$  is the integer from  $\{0, \dots, 255\}$ , while  $D(i, j) \in \{0, \dots, 255\}$ , where  $i = 1, \dots, M$  and  $j = 1, \dots, N$ .

After applying binarization, intensity function is converted into a binary intensity function given by

$$B(i, j) = \begin{cases} 1, & D(i, j) \geq D_{th}(i, j), \\ 0, & D(i, j) < D_{th}(i, j). \end{cases} \quad (1)$$

where  $D_{th}$  is given by any local binarization method [23–26]. It represents local threshold sensitivity decision value. Currently, document image is represented as binary matrix  $\mathbf{B}$  featuring  $M$  rows by  $N$  columns. It consists of only black and white pixels where 0 represents a black pixel and 1 white pixel.

### 2.1 Basic Algorithm

Basic water flow algorithm [21] assumes hypothetical water flows under few angles of the document image frame from left to right and vice versa. In this hypothetically assumed situation, water flows across the image frame

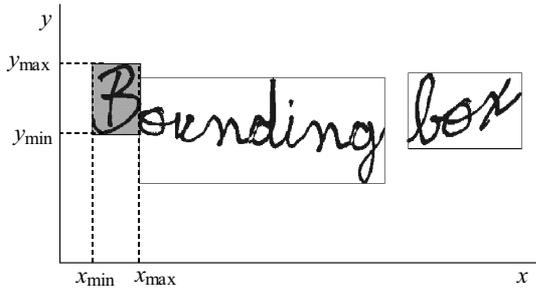


Fig. 4. Bounding box including its endpoints



Fig. 5. Bounding box set over the sample text

	$i$	$i+1$
$j-1$	1	1
$j$	0	1
$j+1$	0	1

(a)

	$i$	$i+1$
$j-1$	0	1
$j$	0	1
$j+1$	1	1

(b)

	$i$	$i+1$
$j-1$	1	1
$j$	0	1
$j+1$	1	1

(c)

Fig. 6. Pixel type determination. (a) — upper boundary pixel, (b) — lower boundary pixel, and (c) — boundary pixel for additional investigation

creating wetted regions. The rest forms unwetted regions. The stripes of unwetted regions are labeled for the extraction of text lines. Further, this hypothetical water flow is expected to fill up the gaps between consecutive text lines. Hence, unwetted regions left on the image frame lies under the text lines. Once the labeling is completed, the image is divided into two different types of stripes. First one contains text lines. The other one contains a space between text lines. Figure 1 shows the hypothetical water that flows from left to right only.

Furthermore, water flow angle  $\alpha$ , as a parameter, is introduced [21]. For the each unwetted region, this angle is measured between  $x$ -axis and the line that endpoint this unwetted region. Nonetheless, it is formed by labeling original document image using a spatial filter mask. These masks have the size  $3 \times P$ , where  $P = \{2, \dots, 5\}$ , [21, 22]. Hence, the algorithm simply creates unwetted regions under fixed flow angles  $\alpha$  from the set  $\{14^\circ, 18^\circ, 26.6^\circ, 45^\circ\}$

[21, 22]. These masks, for the hypothetical water that flows from left to right, are shown in Fig. 2.

However, the algorithm is activated by seed points. These points are defined by the position of the black pixels in the text. If the pixel represents lower or upper boundary one, then the spatial filter mask will be exploited. As the result, unwetted stripes bounded text. Masks application on the text objects is shown in Fig. 3.

## 2.2 Extended Algorithm

In the extended approach, the extraction of the rectangular bounding box over the text objects is its primary task. Consequently, it is a prerequisite for the new approach to water flow algorithm. Hence, bounding boxes represent control areas authorized for the water flow algorithm criteria application. Thus, all text objects are separated before the water flow algorithm application. As a result, the number of touching components after algorithm application is reduced.

The bounding box is a rectangular region whose boundaries are parallel to the coordinate axes. It is defined by its maximum and minimum extents for all axes representing its endpoints:  $x_{\min}$ ,  $y_{\min}$ ,  $x_{\max}$ , and  $y_{\max}$ . Hence, each pixel  $B(i, j)$  that belongs to the box is given by [27]

$$B(i, j) \mid (x_{\min} \leq i \leq x_{\max}) \wedge (y_{\min} \leq j \leq y_{\max}). \quad (2)$$

The illustration of the bounding box as well as its endpoints is shown in Fig. 4.

Inclusion of the point  $B(i, j)$  in a box is tested by verifying four inequalities from (2). If any one of them fails, then the point is not inside the bounding box. After this processing step, all text objects like letters, part of words or words are surrounded by bounding boxes [28]. Bounding box set over the sample text is illustrated in Fig. 5.

However, the algorithm is applied to each text object bounded by its box. It is activated by seed points. These points are defined by its pixel type. The example of the pixel type is shown in Fig. 6.

Due to the pixel type, *ie* upper or lower, slope is  $-\alpha$  or  $+\alpha$ , respectively. However, pixel without complete location has to be additionally investigated. It depends on pixel's neighbor area. Apart from [21], enlarged window composed of  $R \times S$  pixels is determined. In [22]  $R = 5$  and  $S = 7$  is proposed for a complete investigation.

Apart from original algorithm [21], unwetted regions could be determined by lines [22]. Thus, each line is defined as

$$y = ax + b, \quad (3)$$

where slope  $a = \tan \alpha$ . Two lines activated by seed points and defined by angle  $\pm \alpha$  make the connection in specific pixel creating a closed unwetted region [22]. Hence, modifications made on the water flow algorithm is in its formulation. Currently, it creates the water flow function [22] which determines the water flow angle  $\alpha$ . In this way,

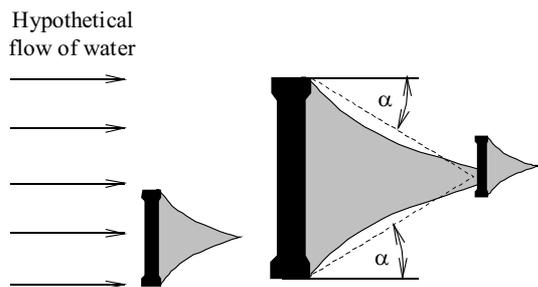


Fig. 7. Unwetted region definition by power water flow function

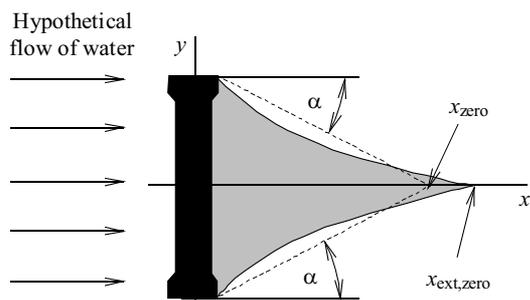


Fig. 8. Function zeroes  $x_{zero}$  and  $x_{ext,zero}$

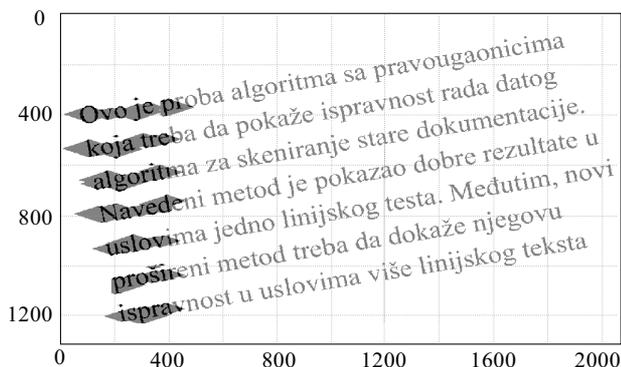


Fig. 9. The process of the water flow algorithm applied to the sample text

making straight lines from boundary pixel and connecting each others in specified point makes the unwetted region.

The principal achievement of unwetted region is text line segmentation. The problem lies in broken words that incorporate in text lines. However, the water flow algorithm should join those words by unwetted regions. These regions can be lengthened by using smaller water angle  $\alpha$  or different water flow function. Hence, modification of the water flow function is proposed [29–30]. From calculus, (3) is a special case of the universal function

$$y = ax^n + b, \tag{4}$$

where (3) and (4) are equal for  $n = 1$ . Using different  $n$  from (4) rebuild different shapes of unwetted regions around the words. Therefore, it forms expanded unwetted region illustrated in Fig. 7.

The water flow function is changed from (3) to (4) by setting  $0 < n \neq 1$ . Basic and expanded water flow function zeroes are marked as  $x_{zero}$  and  $x_{ext,zero}$ , respectively. They are shown in Fig. 8.

Any difference between zeroes leads to the unwetted region extension improving text line segmentation process. Their difference  $x_{diff} = x_{ext,zero} - x_{zero}$  is given by

$$x_{diff} = \left(\frac{H}{2 \tan \alpha}\right)^{1/n} - \left(\frac{H}{2 \tan \alpha}\right), \tag{5}$$

where  $H$  represents the height of the letter "I". Consequently,  $b = H/2$  is valid. If

$$C = \frac{H}{2 \tan \alpha}, \tag{6}$$

then

$$x_{diff} = C^{1/n} - C. \tag{7}$$

The unwetted region is elongated by setting  $x_{diff} > 0$ . Incorporating it in (7) leads to

$$C^{1/n} > C. \tag{8}$$

If parameter  $n < 1$  besides  $C > 0$ , then  $x_{diff} > 0$ . Thus, unwetted region has lengthened as well as expanded. This leads to better text line segmentation.

### 3 EXPERIMENTS

Evaluation of the text line segmentation algorithm is related to its ability to properly perform text line segmentation. Hence, it is performed over different reference samples of text closely related to handwritten text elements as well as the real ones. Experiments framework for the evaluation of algorithm's text line segmentation consists of the following tests [34, 35]:

- multi-line straight text segmentation test,
- multi-line waved text segmentation test,
- multi-line fractured text segmentation test,
- handwritten text segmentation test [36].

#### 3.1 Multi-line Straight Text Segmentation Test

Multi-line straight text segmentation test is based on straight text reference line. Straight text is defined by the skew angle  $\beta$ . Typical values of  $\beta$  that correspond to the handwritten text are those up to  $20^\circ$ . Hence, it takes value from the set  $\{5^\circ, 10^\circ, 15^\circ, 20^\circ\}$  [34, 35]. Furthermore, between line spacing is set to standard values to 20% of the standard character height [37]. This corresponds to single line spacing. Resolution of the text samples is 300 dpi. Multi-line straight text sample is shown in Fig. 10.

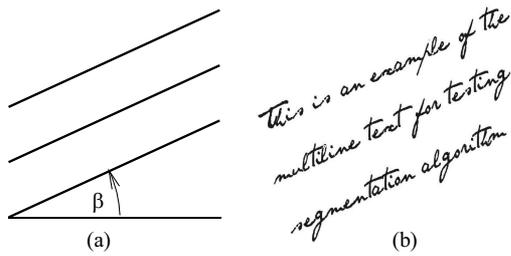


Fig. 10. Multi-line straight text: (a) — reference line definition, (b) — text over reference line

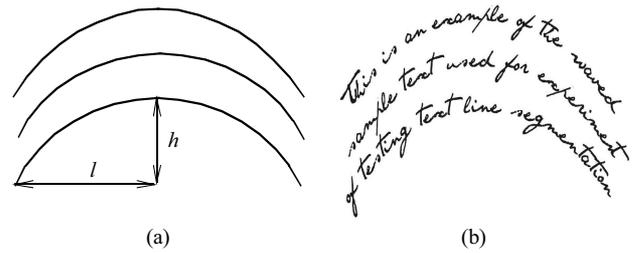


Fig. 11. Multi-line waved text: (a) — reference line definition, (b) — text over reference line

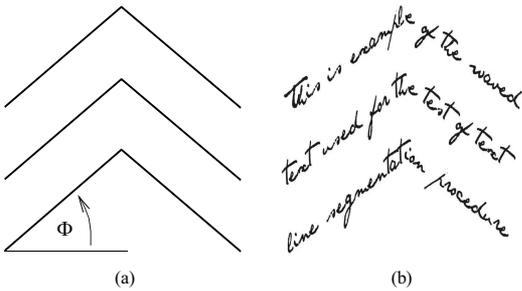


Fig. 12. Multi-line fracture text: (a) — reference line definition, (b) — text over reference line

Радња карактеристика наше и идеје која се  
 управљајуће на системима који управљају и анализу  
 свакодневне радње ситуације. Уколико се у  
 оквиру нашег корпоративног менаџмента,  
 аутоматизацију на разним и одређеним серверима,  
 хардвер и софтверским инсталацијама, подешавања  
 тај је процес извршавајући на менаџменту.

(a)

Design of fractional order controllers based on optimization methods  
 is one of the intensively developed fields at the present time. There  
 are several quality criteria to evaluate the controller performance  
 and to design the controller parameters by optimization. All of these  
 objective functions are almost always multimodal in this case -  
 so they have too complex nature with many local minima.

(b)

Fig. 13. Multi-line handwritten text: (a) — Serbian Cyrillic text, (b) — English text

### 3.2 Multi-line Waved Text Segmentation Test

Multi-line waved text segmentation test is based on waved text reference line. Waved text is determined by

the parameter  $\varepsilon$ . It is given as  $\varepsilon = h/l$ , where  $h$  is height, and  $l$  is half-width of the waved reference line (Fig. 11). Typical values of  $\varepsilon$  that correspond to the chosen values of skew angle  $\beta$  are those from the set  $\{1/12, 1/6, 1/4, 1/3\}$  [34, 35]. Inter-line spacing is set to 20% of the standard character height [37]. The resolution of the text samples is 300 dpi. Multi-line waved text sample is shown in Fig. 11.

### 3.3 Multi-line Fractured Text Segmentation Test

Multi-line fractured text segmentation test is based on fractured text reference line. Fractured text is defined by the fractured skew angle  $\phi$ . Typical values of  $\phi$  that correspond to the handwritten text are those up to  $20^\circ$ . Hence, it receives value from the set  $\{5^\circ, 10^\circ, 15^\circ, 20^\circ\}$  [34, 35]. Furthermore, between line spacing is set to 20% of the standard character height [37]. Resolution of the text samples is 300 dpi. Multi-line fractured text sample is shown in Fig. 12.

### 3.4 Handwritten Text Segmentation Test

Multi-line handwritten text segmentation test is based on freestyle handwritten text samples in Serbian Latin, Cyrillic as well as in English letters [36]. This is a small document text database that consists of 220 text lines. These text samples contain variable skew lines, multi-oriented text as well as mutually inserted the words from different text lines. For the testing purposes document's body is the only considered in the text line segmentation process. Resolution of the text samples is 300 dpi. Few handwritten text fragments from the text database are presented in Fig. 13.

### 3.5 Classification of Text Objects

It is assumed that during test reference sample text containing text objects is processed by algorithm. As a consequence new text objects configuration is obtained. In an ideal circumstance the number of newly arranged objects corresponds to the correct number of text lines. To make a valid algorithm evaluation following text elements should be defined [36]:

- initial objects number  $O_{init}$ ,
- detected objects number  $O_{det}$ , and
- reference objects number  $O_{ref}$ .

Initial objects  $O_{init}$  represent the starting number of objects in the reference sample text. After applying the algorithm over reference sample text counted the number of objects is given as detected objects  $O_{det}$ . Further, the goal *ie* desired number of objects  $O_{ref}$  represents the number of text lines in reference sample text. It is called the reference number of objects. By comparing the reference and detected number of objects the algorithm efficiency is measured.

### 3.6 Classification of Text Segmentation Errors

If the number of text objects in distinct text line is equal to one, then  $O_{det} = O_{ref}$  leading to correct segmented text line. The number of correctly detected text lines in sample text is marked as  $O_{clindet}$ . However, all others are defined as error.

Segmentation errors are present as follows:

- over-segmentation detected text lines  $O_{ovlindet}$  (Split lines error *ie*  $SLE$  [38]),
- under-segmentation detected text lines  $O_{unlindet}$  (Joined lines error, *ie*  $JLE$  [38]), and
- detected text lines with mutually inserted words from different text lines  $O_{mixlindet}$  (Lines including outlier words, *ie*  $LIOW$  [38]).

Split lines error represents the text lines which are wrongly divided by algorithm in two or more components, *ie* text objects. This circumstance is known as over-segmentation. Joined lines error corresponds to the situation where the sequence of  $n$  consecutive lines is considered by the algorithm as a unique line. This phenomenon is called under-segmentation. Lines including outlier words correspond to lines containing words that are incorrectly assigned to two adjacent lines.

### 3.7 Evaluation of Algorithm's Efficiency based on Errors Type

The algorithms efficiency means the evaluation of the text line segmentation process made by investigated algorithm. If the number of detected objects is closer to the number of reference objects, then the algorithm is more efficient. To evaluate the algorithm's efficiency the following elements are introduced:

- segmentation line hit rate, *ie*  $SLHR$ ,
- over-segmentation line hit rate, *ie*  $OSLHR$ ,
- under-segmentation line hit rate, *ie*  $USLHR$ ,
- mixed line hit rate, *ie*  $MLHR$ , and
- segmentation root mean square error, *ie*  $RMSE_{seg}$ .

$SLHR$  represents the ratio of the number of correctly segmented text lines over the total number of text lines in the reference sample text. It is defined as

$$SLHR = 1 - \left| \frac{O_{ref} - O_{clindet}}{O_{ref}} \right|. \quad (9)$$

The over-segmentation phenomena lead to the increased number of objects for text line. Hence, the boundary growing area made by algorithm hasn't been successful in merging all objects of the text into one line. As

previously stated, the number of the over-segmented lines are marked as  $O_{ovlindet}$ .  $OSLHR$  represents the ratio of the number of over-segmented text lines over the total number of text lines in the reference sample text. It is defined as

$$OSLHR = 1 - \left| \frac{O_{ref} - O_{ovlindet}}{O_{ref}} \right|. \quad (10)$$

The under-segmentation process leads to the smaller number of objects than the number of text lines. Hence, two or more consecutive text lines are regarded as a unique one.  $USLHR$  represents the ratio of the number of under-segmented text lines over the total number of text lines in the reference sample text. It is defined as

$$USLHR = 1 - \left| \frac{O_{ref} - O_{unlindet}}{O_{ref}} \right|. \quad (11)$$

The process of mutually injected objects from different text lines leads to the mixed text lines.  $MLHR$  represents the ratio of the number of mixed text lines over the total number of text lines in the reference sample text. It is defined as

$$MLHR = 1 - \left| \frac{O_{ref} - O_{mixlindet}}{O_{ref}} \right|. \quad (12)$$

At the end, the number of detected and reference text object for each text line is compared. Hence, the number of reference text objects per line is equal to 1. The variance evaluation is given  $RMSE$  [34, 35]

$$RMSE_{seg} = \sqrt{\frac{1}{N} \sum_{i=1}^N (O_{i,ref} - O_{i,est})^2}, \quad (13)$$

where  $N$  is the total number of lines in the reference sample text,  $O_{i,ref}$  is the number of reference objects in the text line  $i$  (equal to 1 for each line), and  $O_{i,est}$  is the number of detected objects in the text line  $i$ .

## 4 TESTING RESULTS

Testing is based on text database which consists of 288 lines of straight, waved and fractured text types as well as 216 handwritten text lines. Accordingly, 288 text lines include different scripts like Serbian Cyrillic script, Slavic round Glagolitic script, Bengali script and Serbian Latin script. Furthermore, the handwritten text is based on Serbian Latin and Cyrillic as well as on English scripts. The results for different test types are as follows.

**Table 1.** Results obtained from multi-line straight text segmentation test ( $n = 1$  represents a basic linear function, while  $n \neq 1$  represents a power function)

$\alpha$	10°			12°			14°		
$n$	0.9	0.95	1	0.9	0.95	1	0.9	0.95	1
<i>SLHR</i> (%)	70.83	79.17	87.50	75.00	83.33	70.83	79.17	66.67	62.50
<i>OSLHR</i> (%)	0.00	12.50	12.50	16.67	16.67	29.17	20.83	33.33	37.50
<i>USLHR</i> (%)	29.17	8.33	0.00	8.33	0.00	0.00	0.00	0.00	0.00
<i>MLHR</i> (%)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
<i>RMSE<sub>seg</sub></i>	0.54	0.58	0.50	0.61	0.54	0.65	0.58	0.76	0.79

**Table 2.** Results obtained from multi-line waved text segmentation test ( $n \neq 1$  represents a power function)

$\alpha$	10°			12°			14°		
$n$	0.9	0.95	1	0.9	0.95	1	0.9	0.95	1
<i>SLHR</i> (%)	66.67	75.00	72.92	72.92	68.75	64.58	64.58	68.75	47.92
<i>OSLHR</i> (%)	8.33	12.50	14.58	12.50	20.83	33.33	22.92	27.08	52.08
<i>USLHR</i> (%)	25.00	12.50	12.50	14.58	10.42	2.08	12.50	4.17	0.00
<i>MLHR</i> (%)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
<i>RMSE<sub>seg</sub></i>	0.58	0.50	0.52	0.52	0.56	0.78	0.60	0.61	0.88

**Table 3.** Results obtained from multi-line fractured text segmentation test ( $n \neq 1$  represents a power function)

$\alpha$	10°			12°			14°		
$n$	0.9	0.95	1	0.9	0.95	1	0.9	0.95	1
<i>SLHR</i> (%)	64.58	75.00	87.50	72.92	85.42	85.42	81.25	83.33	77.08
<i>OSLHR</i> (%)	0.00	0.00	2.08	0.00	0.00	6.25	2.08	8.33	20.83
<i>USLHR</i> (%)	35.42	25.00	10.42	27.08	14.58	8.33	16.67	8.33	2.08
<i>MLHR</i> (%)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
<i>RMSE<sub>seg</sub></i>	0.60	0.50	0.35	0.52	0.38	0.38	0.50	0.48	0.69

**Table 4.** Results obtained from multi-line handwritten text segmentation test ( $n \neq 1$  represents a power function)

$\alpha$	10°			12°			14°		
$n$	0.9	0.95	1	0.9	0.95	1	0.9	0.95	1
<i>SLHR</i> (%)	83.33	75.93	61.11	74.07	62.96	38.89	64.81	42.59	35.19
<i>OSLHR</i> (%)	7.41	18.52	33.33	18.52	31.48	55.56	29.63	57.41	64.81
<i>USLHR</i> (%)	3.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
<i>MLHR</i> (%)	5.56	5.56	5.56	7.41	5.56	5.56	5.56	0.00	0.00
<i>RMSE<sub>seg</sub></i>	0.41	0.49	0.97	0.56	0.96	1.43	0.98	1.48	2.11

**Table 5.** *SLHR*(%) for different text types

$\alpha$	10°		12°		14°	
$n$	0.95	1	0.95	1	0.95	1
<i>Straight text V</i>	79.17	87.50	83.33	70.83	66.67	62.50
<i>Waved text</i>	75.00	72.92	68.75	64.58	68.75	47.92
<i>Fractured text</i>	75.00	87.50	85.42	85.42	83.33	77.08
<i>Handwritten text</i>	75.93	61.11	62.96	38.89	42.59	35.19

#### 4.1 Multi-line Straight Text Segmentation Test

Test results obtained from multi-line straight text segmentation test are given in Tab. 1.

From Tab. 1, the incorporation of power function with  $n < 1$ , which defines enlarged unwetted regions, leads to better text line segmentation results given by *SLHR*(%) measure. However, it is not the case for a water flow angle  $\alpha = 10^\circ$ .

#### 4.2 Multi-line Waved Text Segmentation Test

The results of text segmentation for multi-line waved text are given in Tab. 2.

From Tab. 2, the utilization of the  $n < 1$  improves the text line segmentation results given by *SLHR*(%). It is especially expressed for the  $n = 0.95$ . The minimum value of *RMSE<sub>seg</sub>* confirms it.

#### 4.3 Multi-line Fractured Text Segmentation Test

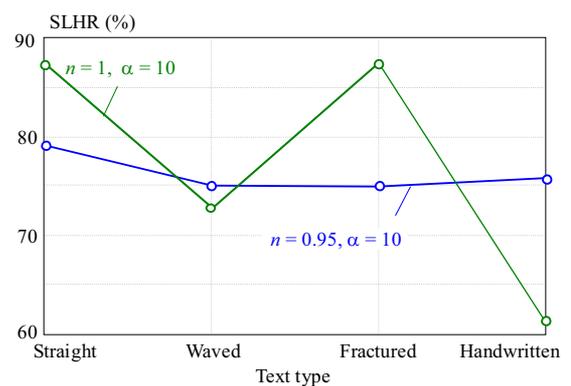
The results of text segmentation obtained from multi-line fractured text are given in Tab. 3.

Again, the incorporation of power function into the algorithm slightly improves the text line segmentation results given by *SLHR*(%). However, it is not the case for the water flow angle  $\alpha = 10^\circ$ .

#### 4.4 Handwritten Text Segmentation Test

The results of text segmentation obtained from handwritten text are given in Tab. 4.

The results from Tab. 4 implicate the advantage of the text line segmentation method based on water flow power function. This is a case by favor of considerably better *SLHR*(%) values for the margin of over 20%. Furthermore, the *RMSE<sub>seg</sub>* follow this trend by obtaining minimal values. As a final note, the test based on multi-line handwritten text is the only real test, while the other three tests are synthetic one. Hence, the results from this test are the key for the algorithm evaluation.



**Fig. 14.** *SLHR* for different text types ( $\alpha = 10^\circ$ ,  $n = \{0.95, 1\}$ )

#### 4.5 Comparative Analysis

At the end, an algorithm based on power function with parameter  $n = 0.95$  gives more uniform text line segmentation results. Hence, the comparison will be made between water flow algorithm with linear and power function, *ie* for  $n = 1$  and  $n = 0.95$ , respectively. The comparison result for *SLHR*(%) is given in Tab. 5.

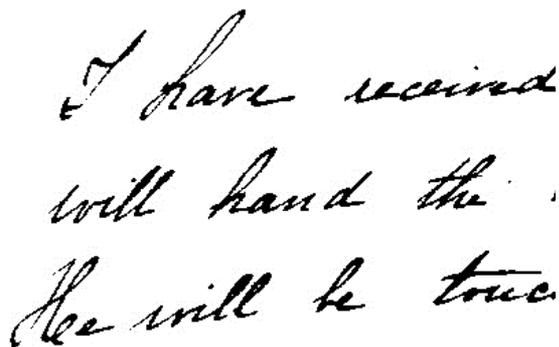


Fig. 15. A sample of the initial binary document image [35]

Table 5 shows that the parameters choice of  $\alpha = 10^\circ$ , and  $n = 0.95$  gives the best as well as the most uniform results. The algorithm that includes this set of parameters is virtually not affected by different text types. Hence, *SLHR* has been always around 75%. The experiments with real text type as handwritten text confirms it. It is illustrated in Fig. 14.

Furthermore, in handwritten text lowering the water flow angle leads to under-segmentation and mixed-segmentation phenomena. The lowering power parameter  $n$  makes the same effect. Hence, this causes the adjacent text lines leading to its merging as well as mixing. In addition, some caution is necessary. To solve this problem, further investigation should point out the improvement of the incorporation of the local skew into the algorithm.

### 5 CONCLUSIONS

The paper presented the water flow algorithm for text line segmentation based on power function. The basic water flow algorithm assumed that hypothetical water flows under few specified angles of the image frame from left to right and vice versa. The algorithm parameter was the water flow angle which established the unwetted regions. These regions were key elements in the text line segmentation process. Aside from basic algorithm, the

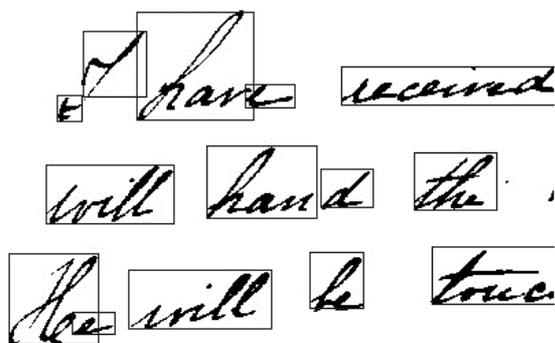


Fig. 17. The bounding box extraction of the connected-components

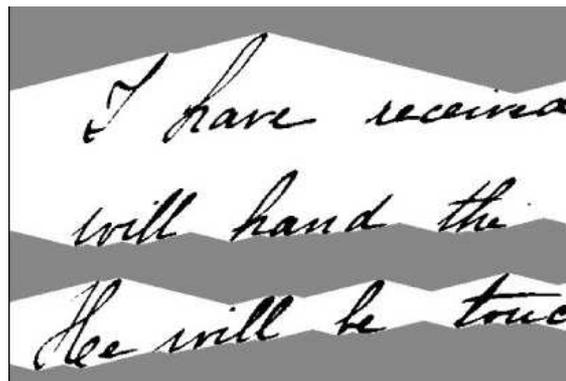


Fig. 16. Water flow algorithm applied to the sample text

introduction of the water flow function instead of a water flow angle gave an opportunity to define new parametric classes of the algorithms. The water flow algorithm based on power function was only one of the many possible. In this version of water flow algorithm, the unwetted regions were defined by the power function  $y = ax^n + b$ . Experiments were made on text database that consist of synthetic as well as the real text samples. Results from experiments and its comparative analysis with the basic water flow algorithm showed the advantage of the extended approach in the area of text line segmentation.

### APPENDIX

To illustrate the complete procedure of the extended water flow algorithm compared to the basic one, the example of the initial binary document image is used. It is shown in Fig. 15.

After that, all versions of the water flow algorithm are applied to that sample text. The resulting document images after the segmentation process are given as follows.

#### Basic (original) water-flow algorithm

First, the basic water flow algorithm is employed. The water flow angle  $\alpha$  is set to  $14^\circ$  [21,35]. The main result is shown in Fig. 16.

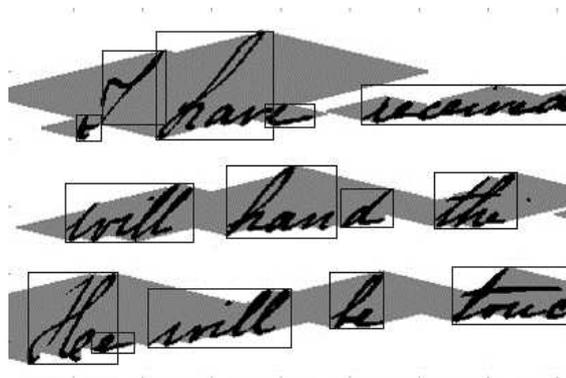


Fig. 18. Water flow algorithm application to the connected-components

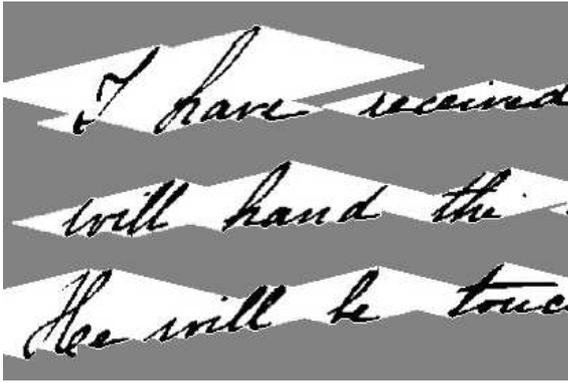


Fig. 19. A water flow algorithm based on linear function applied to the sample text

From Fig. 16 line 1 and 2 are joined leading to the under-segmentation phenomena. Hence, it defines a text segmentation error.

#### Water-flow algorithm based on linear function

In the second case, the extended approach to water flow algorithm for text line segmentation is used. It is based on linear water flow function. The water flow angle  $\alpha$  is set to  $14^\circ$  [42]. In the first step the bounding boxes extracted the connected-components of the initial binary document image. It is shown in Fig. 17. Furthermore, the water flow algorithm based on linear function is applied to the each connected-component.

The main result is shown in Fig. 18. To be consistent the result is presented in Fig. 19 like in Fig. 18.

From Fig. 19 line 1 and 2 are split. Hence, the incorporation of connected-components yield results without under-segmentation phenomena. Yet, the established text line 1 is weak.

#### Water-flow algorithm based on power function

In the third case, the water flow algorithm based on power function is applied. The water flow angle  $\alpha$  is set to  $14^\circ$ , while the power parameter  $n$  is set to 0.9 (if  $n = 1$ , then it is linear function). In the first step the bounding boxes extracted the connected-components of the initial binary document image like in Fig. 18. Furthermore, the water flow algorithm based on power function is applied to the each connected-component.

The result is shown in Fig. 20. From Fig. 20 line 1 and 2 are split. Yet, the established text line 1 is not weak.

#### Acknowledgement

I thank to Prof Zoran N. Milivojević for the invaluable help to greatly improve the manuscript.

This work was partially supported by the Grant of the Ministry of Education, Science and Technological Development of the Republic Serbia, as a part of the project TR33037.

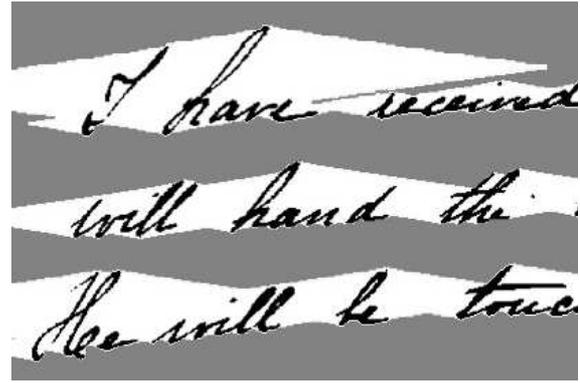


Fig. 20. A water flow algorithm based on power function applied to the sample text

#### REFERENCES

- [1] RAZAK, Z.—ZULKIFLEE, K.*et al*: Off-Line Handwriting Textline Segmentation: A Review, *International Journal of Computer Science and Network Security* **8** No. 7 (2008), 12–20.
- [2] LIKFORMAN-SULEM, L.—ZAHOUR, A.—TACONET, B.: Text Line Segmentation of Historical Documents: A Survey, *International Journal on Document Analysis and Recognition* **9** No. 2–4 (2007), 123–138.
- [3] YI, L.—YEFENG, Z.—DOERMANN, D.—JAEGER, S.: Script-Independent Text Line Segmentation in Freestyle Handwritten Documents, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **30** No. 8 (2008), 1313–1329.
- [4] PLAMONDON, R.—SRIHARI, S. N.: Online and Offline Handwriting Recognition: A Comprehensive Survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22** No. 1 (2000), 63–84.
- [5] ZRAMDINI, A.—INGLUD, R.: Optical Font Recognition from Projection Profiles, *Electronic Publishing* **6** No. 3 (1993), 249–260.
- [6] SILVA, L. F.—CONCI, A.—SANCHEZ, A.: Automatic Discrimination between Printed and Handwritten Text in Documents, In *Proceedings of 22nd Brazilian Symposium on Computer Graphics and Image Processing*, Rio de Janeiro (Brazil), 2009, pp. 261–267.
- [7] BALLARD, D. H.: Generalizing the Hough Transforms to Detect Arbitrary Shapes, *Pattern Recognition* **13** No. 2 (1981), 111–122.
- [8] AMIN, A.—FISCHER, A. A.: Document Skew Detection Method using the Hough Transform, *Pattern Analysis & Applications* **3** No. 3 (2000), 243–253.
- [9] BALLARD, D. H.—BROWN, C. M.: *Computer Vision*, Prentice Hall, Englewood Cliffs, N.J., U.S.A., 1982, pp. 149–165.
- [10] WAHL, F. M.—WONG, K. Y.—CASEY, R. G.: Block Segmentation and Text Extraction in Mixed Text/Image Documents, *Computer Graphics and Image Processing* **20** No. 2 (1982), 375–390.
- [11] SHI, Z.—GOVINDARAJU, V.: Line Separation for Complex Document Images using Fuzzy Runlength, In *Proceedings of the International Workshop on Document Image Analysis for Libraries '04*, Palo Alto (U.S.A.), 2004, pp. 306–312.
- [12] YIN, F.—LIU, C.: A Variational Bayes Method for Handwritten Text Line Segmentation, In *Proceedings of the 10th International Conference on Document Analysis and Recognition (ICDAR '09)*, Barcelona (Spain), 2009, pp. 436–440.
- [13] TRIPATHY, N.—PAL, U. Handwriting Segmentation of Unconstrained Oriya Text: *Sadhana* **31** No. 6 (2006), 755–769.

- [14] DU, X.—PAN, W.—BUI, T. D.: Text Line Segmentation in Handwritten Documents using Mumford-Shah Model, *Pattern Recognition* **42** No. 12 (2009), 3136–3145.
- [15] LIKFORMAN-SULEM, L.—FAURE, C. Extracting Lines on Handwritten Documents by Perceptual Grouping: In *Advances in Handwriting and Drawing: a Multidisciplinary Approach* (C. Faure, P. Keuss, G. Lorette, A. Winter, eds.), Paris: Europia, 1994, pp. 21–38.
- [16] KHANDEWAL, A.—CHOUDHURY, P.—SARKAR, R.—BASU, S.—NASIPURI, M.—DAS, N.: Text Line Segmentation for Unconstrained Handwritten Document Images using Neighborhood Connected Component Analysis, *Proceedings of the 3rd International Conference on Pattern Recognition and Machine Intelligence (PReMI '09)* (S. Chaudhury *et al.*, eds.), LNCS 5909, pp. 369–374.
- [17] ZAHOUR, A.—TACONET, B.—MERCY, P.—RAMDANE, S.: Arabic Handwritten Text-Line Extraction, In *Proceedings of the 6th International Conference on Document Analysis and Recognition (ICDAR '01)*, Seattle (U.S.A.), 2001, pp. 281–285.
- [18] KOSHINAKA, T.—KEN'ICHI, I.—AKITOSHI, O.: An HMM-Based Text Segmentation Method using Variational Bayes Approach, *IEIC Technical Report* **104** No. 87 (2004), 19–24.
- [19] ABUHAIBA, I. S. I.—DATTA, S.—HOLT, M. J. J.: Line Extraction and Stroke Ordering of Text Pages, In *Proceedings of the 3rd International Conference on Document Analysis and Recognition (ICDAR '95)*, Montreal (Canada), 1995, pp. 390–393.
- [20] SESH KUMAR, K. S.—NAMBOODIRI, A. M.—JAWAHAR, C. V.: Learning Segmentation of Documents with Complex Scripts, In *Proceedings of the 5th Indian Conference on Computer Vision, Graphics and Image Processing*, LNCS 4338, Madurai (India), 2006, pp. 749–760.
- [21] BASU, S.—CHAUDHURI, C.—KUNDU, M.—NASIPURI, M.—BASU, D. K.: Text Line Extraction from Multi-Skewed Handwritten Document, *Pattern Recognition* **40** No. 6 (2006), 1825–1839.
- [22] BRODIĆ, D.—MILIVOJEVIĆ, Z.: A New Approach to Water Flow Algorithm for Text Line Segmentation, *Journal of Universal Computer Science* **17** No. 1 (2011), 30–47.
- [23] NIBLACK, W.: *An Introduction to Digital Image Processing*, Prentice Hall, Englewood Cliffs, N.J., U.S.A., 1986.
- [24] SAUVOLA, L.—PIETIKAINEN, M.: Adaptive Document Image Binarization, *Pattern Recognition* **33** No. 2 (2000), 225–236.
- [25] STATHIS, P.—KAVALLIERATOU, E.—PAPAMARKOS, N.: An Evaluation Technique for Binarization Algorithms, *Journal of Universal Computer Science* **14** No. 18 (2008), 3011–3030.
- [26] KHASMAN, A.—SEKEROGLU, B.: Document Image Binarization using a Supervised Neural Network, *International Journal of Neural Systems* **18** No. 5 (2008), 405–418.
- [27] PREPARATA, F. P.—SHAMOS, M. I.: *Computational Geometry: An Introduction*, Springer, Berlin, 1995.
- [28] WANG, J.—LEUNG, M. K. H.—HUI, S. C.: Cursive Word Reference Line Detection, *Pattern Recognition* **30** No. 3 (1997), 503–511.
- [29] BRODIĆ, D.—MILIVOJEVIĆ, Z.: An Approach to Modification of Water Flow Algorithm for Segmentation and Text Parameters Extraction, In *Emerging Trends in Technological Innovation* (L.M. Camarinha-Matos, P. Pereira, L. Ribeiro, eds.), *Proceedings of First IFIP WG 5.5/SOCOLNET Doctoral Conference on Computing, Electrical and Industrial Systems (DOCEIS '2010)*, vol. 314, Costa de Caparica (Portugal) IFIP AICT, pp. 324–331, DOI: 10.1007/978-3-642-11628-5\_35.
- [30] BRODIĆ, D.—MILIVOJEVIĆ, Z.: Text Line Segmentation by Adapted Water Flow Algorithm, In *Proceedings of 10th Symposium on Neural Network Applications in Electrical Engineering (NEUREL '2010)*, Belgrade (Serbia), 2010, pp. 225–229.
- [31] BRODIĆ, D.—MILIVOJEVIĆ, D. R.—MILIVOJEVIĆ, Z.: Basic Test Framework for the Evaluation of Text Line Segmentation and Text Parameter Extraction, *Sensors* **10** No. 5 (2010), 5263–5279.
- [32] BRODIĆ, D.: Basic Experiments Set for the Evaluation of the Text Line Segmentation, *Przeglad Elektrotechniczny (Electrical Review)* **86** No. 11 (2010), 353–357.
- [33] BRODIĆ, D.: Advantages of the Extended Water Flow Algorithm for Handwritten Text Line Segmentation, In *Proceedings of the 4th International Conference Pattern Recognition and Machine Intelligence (PReMI '11)*, Moscow (Russia), 2011, (accepted).
- [34] BRODIĆ, D.—DOKIĆ, B.: Initial Skew Rate Detection using Rectangular Hull Gravity Center, In *14th International Conference on Electronics – E'2010*, Vilnius (Lithuania), 2010 Sect.3.24, pp. 1–6.
- [35] SANCHEZ, A.—SUAREZ, P. D.—MELLO, C. A. B.—OLIVEIRA, A. L. I.—ALVES, V. M. O.: Text Line Segmentation in Images of Handwritten Historical Documents, In *Proceedings of First International Workshops on Image Processing Theory, Tools and Applications*, Sousse (Tunisia), 2008, pp. 1–6.
- [36] MING, M.—PENG, Y.—SPRING, M.: Ontology Mapping: As a Binary Classification Problem, In *Proceedings of the 4th International Conference on Semantics, Knowledge and Grid*, Beijing (China), 2008.
- [37] SWETS, J. A.: Measuring the Accuracy of Diagnostic Systems, *Science (New Series)* **240** No. 4857 (1988), 1285–1293.
- [38] FAWCETT, T.: ROC Graphs: Notes and Practical Considerations for Data Mining Researchers, *HP Invent* (2003), 1–27.
- [39] ABDI, H.: Signal Detection Theory, In *Encyclopedia of Measurement and Statistics* (Salkind, ed.), 2007, pp. 1–9.
- [40] QIAN, X.—LIU, G.—WANG, H.—SU, R.: Text Detection, Localization, and Tracking in Compressed Video, *Signal Processing: Image Communication* **22** No. 9 (2007), 752–768.
- [41] BUKHARI, S. S.—SHAFAIT, F.—BRUESL, T. M.: Adaptive Binarization of Unconstrained Hand-Held Camera-Captured Document Images, *Journal of Universal Computer Science* **15** No. 18 (2009), 3343–3363.
- [42] BRODIĆ, D.—MILIVOJEVIĆ, Z.: A New Approach to Water Flow Algorithm for Text Line Segmentation, *Journal of Universal Computer Science* **17** No. 1 (2011), 30–47.
- [43] BRODIĆ, D.—MILIVOJEVIĆ, D. R.—MILIVOJEVIĆ, Z.: An Approach to a Comprehensive Test Framework for Analysis and Evaluation of Text Line Segmentation Algorithms, *Sensors* **11** No. 9 (2011), 8782–8812.

Received 6 February 2014

**Darko Brodić** (Prof, PhD) was born in Konjic, BiH, Yugoslavia, in 1963. He received his BEE and MEE from the Faculty of Electrical Engineering, University of Sarajevo in 1987 and 1990, respectively, as well as PhD from the Faculty of Electrical Engineering, University of Banja Luka in 2011. Currently, he is with the University of Belgrade, Technical Faculty in Bor. His research interests include document image processing. He is the author and coauthor of over 60 journal (over 25 in journals indexed by Thomson JCR) and conference papers as well as few books and the textbooks.