General Character Segmentation Approach for Machine-Typed Documents

Vladan Vučković, Boban Arizanović

***Abstract*—In this paper one important entirety in most OCR systems is discussed. The main goal is to explain the approach for segmentation of the individual characters from machine-typed documents, which is based on different image processing techniques in spatial domain and exploits the features of a document structure. The paper gives the overview of the proposed technique by the phases, starting from the grayscale conversion and binarization, through the line, word, and character segmentation of a document image using the modified projection profiles technique, as well as the description of the problems which can appear. The approach is semi-automatic since it requires adjustment of the thresholds for the different segmentation parameters. The numerical results which show the proposed approach performances from the perspective of the time complexity and segmentation accuracy are provided. Results show that this approach outperforms the state-of-the-art methods in all aspects. Furthermore, it is shown that the proposed character segmentation technique is suitable for real-time tasks.**

***Index Terms*—Image Processing, OCR, Character Segmentation, Machine-Typed Documents.**

I. Introduction

Character segmentation still represents very challenging task in image processing and other related fields. It is a crucial pre-processing part in most OCR systems [1,2], and combined with character recognition [3] represents an important subject of research for a long period of time [4]. The process of character segmentation is usually unduly underestimated comparing to the process of character recognition [5,6]. Some approaches deal with extraction of characters from natural and other non-document images [7,8], and others deal with character segmentation on document images. The second group is usually divided into machine-printed documents [5,9,10,11], where the shape of document elements is regular, and handwritten documents where character segmentation is challenged due to irregular document structure [6,12,13,14,15]. Machine-typed documents are especially important because of historical documents [3,16,17].

All stages in character segmentation have been a subject of research. One of the most important problems which can appear in the process of character segmentation is a document skew and a variety of works deal with this problem. Many skew estimation approaches are modifications of the Hough transform [18,19], and some of them are based on correlation functions or straight line fitting [20,21]. Analyses on image binarization parameters proved the superiority of Otsu method and other Otsu-based methods [22]. Gaussian low-pass filter and innovational Laplace-like transform can be used for character segmentation [23]. Segmentation can be also performed using the Bayes theorem, in order to ensure the usage of the prior knowledge [24]. One character segmentation approach for typewritten historical documents is based on the usage of the horizontal projection profile of each word segment [16]. Adaptive run length smoothing (ARLSA) algorithm for segmentation of historical machine-printed documents proved to be better than state-of-the-art methods [10]. Diverse approaches for segmentation of handwritten documents are proposed [13]. A clustering based method can be exploited in the process of segmentation [25]. Gabor filter can be exploited for feature extraction and Fisher classifier for feature classification [26]. In order to solve the problem with touching characters, self-organizing maps, SVM classifiers, and Multi-Layer Perceptron are used [12,14,27]. For natural images, some approaches are based on tensor voting and on the usage of the three-color bar code for segmentation [8,28].

This paper presents general character segmentation approach for machine-typed documents and provides experimental results for time complexity and segmentation accuracy. This technique can be also used for processing of machine-printed documents. The proposed approach uses the modified projection profiles technique which exploits the spatial features of a document structure in the segmentation process. Provided numerical results show that this approach generally gives better results than state-of-the-art methods. For the purpose of testing the approach performances, historical ground-truth machine-printed documents are used. It should be mentioned that this approach is designed as a part of a real-time OCR system for the needs of the "Nikola Tesla Museum" [29].

This paper is organized as follows: in section 2 the complete representation of the proposed character segmentation approach is provided. In section 3 the approach performances are analyzed using the set of ground-truth historical machine-printed documents and comparison with the state-of-the-art approaches is given. Finally, the discussion about the performances and the future work is given in section 4.

II. Character Segmentation Approach

The approach consists of several steps and each step is explained in details. Image processing techniques in spatial domain for image binarization and noise reduction [30,31] and histogram processing methods [15,16,32], are used in the development of the proposed approach. The original

Vladan Vučković is with the Faculty of Electronic Engineering, University of Niš, 14 Aleksandra Medvedeva, 18000 Niš, Serbia (e-mail: vladanvuckovic24@gmail).

Boban Arizanović is with the Faculty of Electronic Engineering, University of Niš, 14 Aleksandra Medvedeva, 18000 Niš, Serbia (e-mail: bobanarizanovic@hotmail.com).

document image used for demonstration of the proposed approach steps is shown in Fig. 1.

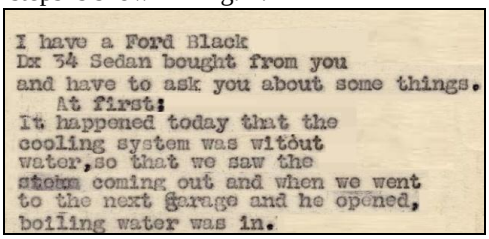

Fig. 1. Original image.

Original document image shown in Fig. 1. represents an example of machine-typed documents which the presented technique is designed for.

2.1 Grayscale conversion

In order to obtain a binarized document image, the simplest way for conversion of the color image to grayscale image is exploited (1). Suppose that *f* is the starting 24-bit image, size of M x N, and *g* is 8-bit image of the same size. Each pixel g(x,y) takes the following intensity value:

$$g(x,y) = \max(f(x,y)_R, f(x,y)_G, f(x,y)_B) \quad (1)$$

for x = 0, 1, 2,…, M – 1 and y = 0, 1, 2,…, N – 1, while *max* determines the maximum of three components of the pixel intensity value. A grayscale image obtained in this step is shown in Fig. 2.

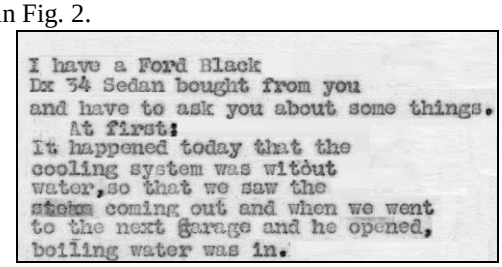

Fig. 2. Grayscale image.

After the grayscale conversion, obtained document image contains only the gray level pixel intensity values which is necessary for the process of binarization.

2.2 Binarization

This part performs conversion from grayscale to binary image using a thresholding function (2). Suppose that input to this pre-processing part is 8-bit grayscale image *f* with pixel intensity values in range 0-255. Also, suppose that the threshold value used for obtaining the pixel intensity values in resulting image is equal to T_{hb}. Each pixel in image *g* will take the value:

$$T(r) = \begin{cases} 1, & 0 \le r \le T_{hb} \\ 0, & T_{hb} < r \le r_{\max} \end{cases} \quad (2)$$

for x = 0, 1, 2,…, M – 1 and y = 0, 1, 2,…, N – 1. Thresholding function used in this process is shown in Fig. 3.

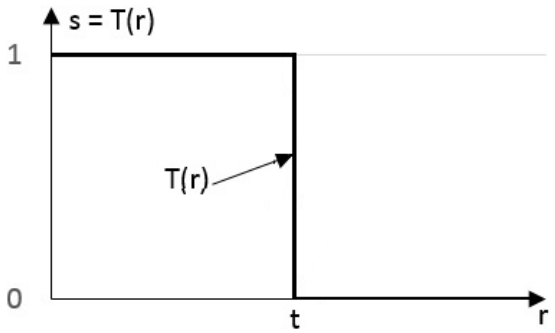

Fig. 3. Thresholding function.

Thresholding value that gives the best results depends primarily on the quality of a document image.

2.3 Noise reduction

After the image binarization usually there is a problem with isolated groups of black pixels which represent the noise and are not desirable for further processing. This part of the pre-processing stage performs simple reduction of the isolated black pixels which do not have other black pixels as 8-neighbors (3). Suppose that *f* is a binary image, size of (M+2) x (N+2), obtained after the binarization and expanded with zeros on all four sides due to border processing. Suppose that w is a filter mask size of m x n, where m = 2a + 1 and n = 2b + 1, where a and b are positive integer values. Each pixel in the image *g* size of M x N will have the following intensity value:

$$g(x,y) = \begin{cases} f(x,y), & \sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t) f(x+s,y+t) > 0 \\ 0, & \sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t) f(x+s,y+t) = 0 \end{cases} \quad (3)$$

In this case the filter mask w is 3x3 in size with 1s as edge elements, where a and b are also equal to 1, noting that the middle element of the filter mask is w(0, 0). The filter mask is shown in Fig. 4.

1	1	1
1	0	1
1	1	1

Fig. 4. 3x3 filter mask used for noise reduction.

The filter mask shown in Fig. 4 is chosen because it does not make any damage to the important document image features. A binary image obtained after the image binarization followed by noise reduction is shown in Fig. 5.

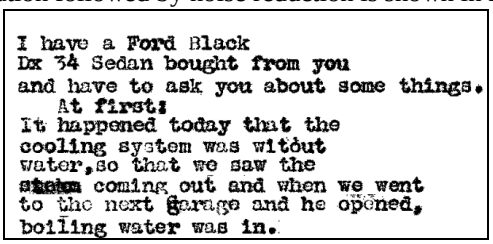

Fig. 5. Binary image.

The combination of binarization and noise reduction proved to be very effective since the resulting image is filtered and quality of the characters remained at a high level.

2.4 Line segmentation

One of the central tasks which the proposed approach must perform is the line segmentation of a document image. The method used here is based on projection profiles technique. Suppose that the sliding window is size of W x N, where W is the height of the sliding window and N is the width of the binary image *f*. This process is described as follows:

Step 1: The sliding window is positioned on the top of image *f* and moved down the vertical axis by 1 pixel. For each moving of the sliding window the sum of all black pixels in window is calculated as in (4).

$$s_n = \sum_{s=0}^{H-1}\sum_{y=0}^{N-1} f(x+s,y) \quad (4)$$

Step 2: If $s_n < T_{hswl}$, all pixels in the middle line of the sliding window will be taken as delimiters between the document lines. That pixel scanline will have the offset value x + W/2 relative to the top of the image, where W/2 represents integer division.

Step 3: The borders of the potential document lines are obtained by performing a top down image analysis, i.e. by analyzing the offsets obtained in the previous step. Using this offset analysis, the closest offsets on distance greater than 1 are determined. Processing is performed starting from the first offset. Once the next offset on distance greater than 1 is found, both offsets are taken as upper and lower borders of the document line. The process is repeated until the last offset.

Step 4: Since the document line height is greater than d pixels, it is necessary to eliminate the segments that have a distance between the upper and the lower border less than value d. The best choice for value d is to take the approximate value of the document character height. At the end of this step, the segments which represent the line borders of the document image are obtained and are taken as input for the next stage.

The image obtained after the line segmentation is shown in Fig. 6.

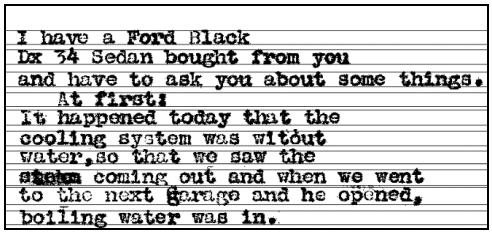

Fig. 6. Line segmentation results.

This process is one of the most important parts of the proposed approach and highly affects the further processing.

2.5 Word segmentation

The next stage is focused on analysis of each detected document line and segmentation of each line into words. This stage also includes the detection of the punctuation characters, some of which could be even recognized based on the concentration of black pixels in specific area of the line. Steps performed in this part are the following:

Step 1: Similar as in the previous stage, the sliding window size of W x H is taken, where H represents the height of the current line, $H = L_{cu} - L_{cl} + 1$, where L_{cu} and L_{cl} are offsets of the upper and lower border of the current line, respectively, and starting from the top left position of the current line the number of black pixels in the sliding window is calculated as in (5).

$$h_n = \sum_{x=L_{cu}}^{L_{cl}} \sum_{t=0}^{W-1} f(x, y+t) \tag{5}$$

where *f* represents a binary image.

Step 2: The window is sliding 1 pixel to the right along the horizontal axis and all values h_n are obtained.

Step 3: Using array H of values h_n, the histogram is computed for each document line.

Step 4: The key of the histogram analysis is determination of the local minima which represent the biggest downs of the values, in this case the biggest downs of the concentration of black pixels in a given line. The local minima less than thresholding value t are taken. These values represent delimiters between the words.

First line of a document image used for approach representation is shown in Fig. 7.

Fig. 7. First line of a document image.

The corresponding computed histogram for obtained document line from Fig. 7 is shown in Fig. 8.

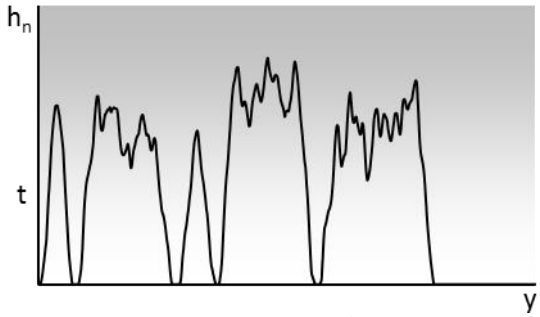

Fig. 8. Corresponding computed histogram for obtained document line.

Finally, the resulting image obtained after the word segmentation is shown in Fig. 9.

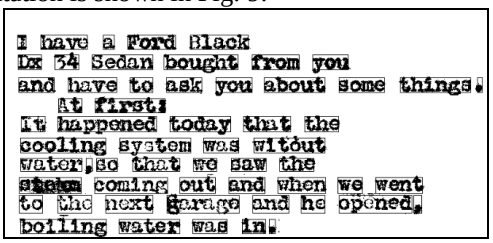

Fig. 9. Word segmentation results.

The word segmentation results primarily depend on spaces between the words and characters. This will be discussed in the experimental section.

2.6 Character segmentation

Once the word segmentation is performed, it is followed by character segmentation. The approach used here is based on combination of standard projection profiles technique and proposed decision-making logic [29]. The first part of character segmentation is a standard segmentation based on histogram processing. In this case the histogram minima represent the potential delimiters between characters. Since there are many delimiters which are considered as potential ones, it is necessary to make decision and chose the ones which are the most likely to be the real delimiters. The decision-making logic used for determining the real delimiters is shown in Fig. 10.

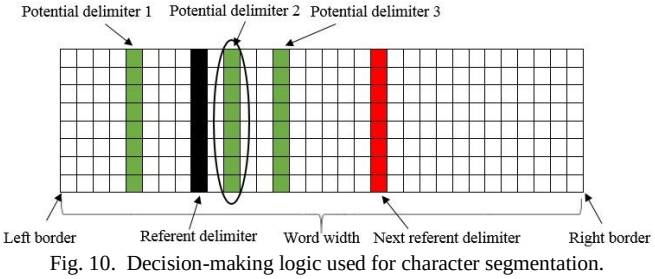

Fig. 10. Decision-making logic used for character segmentation.

In order to eliminate the possible segmentation errors, the word width and threshold value for average character width are used. Using these values, the number of characters for a given word is calculated as in (6).

$$c_n = \left\lfloor \frac{W_w}{T_{hcw}} \right\rfloor \tag{6}$$

where W_w is a word width in pixels and T_{hcw} is assumed average character width in a document image. Finally, the

actual average character width can be calculated using the determined number of characters in a given word as in (7).

$$c_{wavg} = \left\lfloor \frac{W_w}{c_n} \right\rfloor \quad (7)$$

The given word is separated starting from the left border and taking the delimiter at distance c_{wavg} as a referent delimiter. The real delimiter is determined by finding the closest delimiter to the referent delimiter in range controlled by the threshold value as in (8) and (9). If there are no delimiters in that range, the referent delimiter will be taken as a real delimiter.

$$j = \arg\min_i(|d_i - d_{ref}|) \quad (8)$$

$$d = \begin{cases} d_j, |d_j - d_{ref}| \le T_{hoffset} \\ d_{ref}, |d_j - d_{ref}| > T_{hoffset} \end{cases} \quad (9)$$

where d_i and d_{ref} are current and referent delimiter, respectively, and $T_{hoffset}$ is a threshold value which controls the segmentation. The pseudo-code for the proposed decision-making logic is shown in the following listing.

```
1:  for each word W in WORDS do
2:  D ← Ø
3:  CHARACTERS ← Ø
4:  CHARACTERS ← SLIDING-WINDOW-METHOD(W)
5:  W_L = W_w div T_hcw
6:  C_wavg = W_w div W_L
7:  D_ref = START-POSITION(W) + C_wavg
8:  while COUNT(D) ≠ W_L - 1 do
9:  J = FIND-CLOSEST-DELIMITER-POSITION (CHARACTERS, D_ref)
10: if ABS(CHARACTERS [J] - D_ref) < T_hoffset then
11: D ← CHARACTERS [J]
12: D_ref = CHARACTERS [J] + C_wavg
13: else
14: D ← D_ref
15  D_ref = D_ref + C_wavg
16: endif
17: end while
18: DELIMITERS ← D
19: end for
```

The sign "←" in pseudo-code represents the assignment operator for list.

In the case of punctuation marks, the position of black pixels inside the sliding window is considered, e.g. in case of dots and commas the important fact is that most of black pixels are concentrated in the lower half of the line and in case of dashes black pixels are concentrated in the second third of the line. The final resulting image obtained as the output of this stage is shown in Fig. 11.

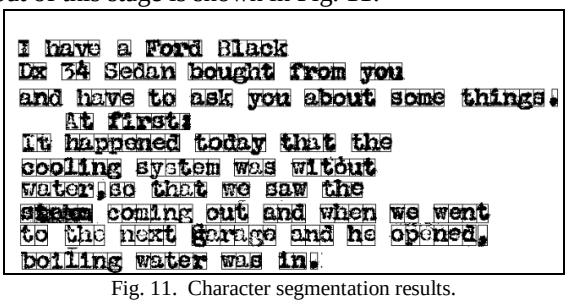

Fig. 11. Character segmentation results.

The output image shows that in this case the proposed decision-making logic provides very accurate segmentation.

3 Experiments

Historical ground-truth machine-printed documents are used for testing the performances of the proposed approach. For the purpose of testing the approach performances, PC machine with an AMD Quad Core Processor running at 3.1 GHz, 4 GB RAM installed, and running operating system Windows 8.1 is used.

A couple of state-of-the-art methods are used for comparison with the proposed approach. Beside the standard projection profiles based approach, also the run length smearing (RLSA) based approach is implemented and comparative results are provided. RLSA technique is based on conversion of the white pixel runs to black pixel runs for each scanline, if runs length is below the threshold value. RLSA technique can be applied horizontally and vertically, the same as projection profiles technique. It can also be used in combination with the projection profiles technique. Taking the second case into consideration, RLSA has a goal to strengthen the histogram by elimination of the local minima with a low possibility of becoming the delimiters. The commercial product FineReader 12 [34] and the Open Source OCROpus software [35] are also used for comparison with the proposed character segmentation approach.

Comparative results are obtained using the manually chosen ground-truth sets of document images for each segmentation level. For text line segmentation 74 images (4363 text line segments), for word segmentation 58 images (20584 word segments), and for character segmentation 22 images (34623 character segments) are chosen.

The matching score metric is used for evaluation of the approach performances [10,33]. This metric represents the ratio between the number of pixels which belong to the segmented text line, word, or character region in document image processed by the proposed approach and ground-truth image, and total number of pixels in the given segmented region in the ground-truth image. In order to control the evaluation criteria, the threshold acceptance value for matching score is used. The higher threshold value means the more rigid evaluation criteria. Comparative results for all segmentation levels, when acceptance threshold is set to 90, are shown in Table 1.

TABLE I
COMPARISON OF THE SEGMENTATION RESULTS FOR DIFFERENT APPROACHES USING THE CHOSEN SETS OF THE GROUND-TRUTH DOCUMENT IMAGES

	Detection rate (%)		
	Text line segmentation	Word segmentation	Character segmentation
Projection profiles based approach	73.39	71.64	72.56
RLSA based approach	72.57	74.85	72.36
FineReader	69.35	76.18	64.82
OCROpus	75.18	79.48	74.43
Proposed approach	**80.58**	**77.57**	**86.14**

Based on obtained results for segmentation accuracy, the proposed approach, in most cases, outperforms the state-of-the-art methods. In the case of the text line segmentation, the advantage of the proposed approach over the standard projection profiles based approach, which uses the projection profiles technique on all levels of segmentation, lies in fact that proposed approach uses a few corrections

applied on obtained document lines. The proposed approach performs better when it comes to word segmentation, since it uses more adaptive algorithm than standard projection profiles technique. Character segmentation using the proposed approach performs much better than state-of-the-art approaches, since the proposed approach uses the proposed decision-making logic which gradually eliminates the possibility of big segmentation errors.

Further evaluation of the proposed approach performances is performed by obtaining the segmentation results for different categories of segmentation problems. Documents used for this evaluation are multi column documents, noisy documents, documents with non-constant spaces between text lines, words, and characters, documents with various font sizes, and warped and/or skewed documents. Detection rate for different character segmentation approaches regarding the different categories of segmentation problems are shown in Tables 2-6.

TABLE 2
SEGMENTATION RESULTS FOR DIFFERENT CATEGORIES OF SEGMENTATION PROBLEMS (PROJECTION PROFILES BASED APPROACH)

	Multi column	Noisy	Non-constant spaces	Various font sizes	Warped-skewed text
Text line segmentation	70.59	67.29	68.41	52.38	59.30
Word segmentation	66.37	58.14	62.67	58.62	55.23
Character segmentation	72.84	65.93	70.28	71.40	67.91

TABLE 3
SEGMENTATION RESULTS FOR DIFFERENT CATEGORIES OF SEGMENTATION PROBLEMS (RLSA BASED APPROACH)

	Multi column	Noisy	Non-constant spaces	Various font sizes	Warped-skewed text
Text line segmentation	69.21	50.72	67.32	53.87	66.74
Word segmentation	68.17	52.73	64.71	62.44	62.39
Character segmentation	74.59	63.54	68.29	75.36	75.97

TABLE 4
SEGMENTATION RESULTS FOR DIFFERENT CATEGORIES OF SEGMENTATION PROBLEMS (FINEREADER)

	Multi column	Noisy	Non-constant spaces	Various font sizes	Warped-skewed text
Text line segmentation	67.55	55.63	59.31	70.52	57.43
Word segmentation	68.15	53.76	56.46	69.49	55.81
Character segmentation	70.26	44.03	58.39	70.34	64.73

TABLE 5
SEGMENTATION RESULTS FOR DIFFERENT CATEGORIES OF SEGMENTATION PROBLEMS (OCROPUS)

	Multi column	Noisy	Non-constant spaces	Various font sizes	Warped-skewed text
Text line segmentation	71.13	73.86	68.44	69.81	62.65
Word segmentation	80.43	68.72	69.39	75.37	74.45
Character segmentation	72.67	64.66	66.60	72.56	73.67

Regarding the text line segmentation, the proposed approach gives better results than all state-of-the-art approaches. Segmentation problems can appear in case of skewed documents. The same type of documents can cause the problems with word segmentation, which can also have problems with noisy documents, but the results are good in most of the cases. The proposed decision-making logic used for character segmentation gives the best results due to the nature of the segmentation logic.

TABLE 6
SEGMENTATION RESULTS FOR DIFFERENT CATEGORIES OF SEGMENTATION PROBLEMS (PROPOSED APPROACH)

	Multi column	Noisy	Non-constant spaces	Various font sizes	Warped-skewed text
Text line segmentation	77.46	75.56	72.34	71.95	70.69
Word segmentation	75.24	71.49	73.37	68.64	65.36
Character segmentation	86.71	80.38	87.06	87.26	82.10

When it comes to the state-of-the-art approaches, projection profiles based approach is mainly affected by noisy documents. Significant document skew can also cause the bad segmentation results. RLSA based approach has problems with noisy documents and documents with non-constant spaces. FineReader technique performs badly in case of noisy documents, while OCROpus technique provides the worst results with documents containing non-constant spaces.

Results from the aspect of the time complexity are also provided. The proposed approach is intended for use in real-time OCR system and this is the main reason for choosing the projection profiles technique, which proved to be quite efficient. The projection profiles technique is used on all levels of segmentation and its processing time determines the overall processing time of the proposed approach. The comparison of the processing time for different character segmentation approaches is given in Table 7.

TABLE 7
COMPARISON OF THE SEGMENTATION PROCESSING TIME FOR DIFFERENT APPROACHES

	Processing time (ms)		
	Projection profiles based approach	RLSA based approach	Proposed approach
1736x4872			
18 text lines	55.84976	128.36514	55.53348
29 text lines	89.12796	209.14872	88.03042
42 text lines	123.55864	304.77813	122.10422
56 text lines	164.13659	365.12239	162.74385
64 text lines	193.28441	405.28846	187.69245
5072x4312			
16 text lines	102.21373	233.44023	101.77729
25 text lines	181.36465	422.36867	180.22220
42 text lines	326.07428	736.58552	324.41320
50 text lines	377.49667	821.27488	374.59168

Comparative results for processing time prove that the proposed approach is more efficient than state-of-the-art methods. Since the text line segmentation is performed at the start of segmentation process, the results are provided for documents with different number of text lines. Based on the results. the proposed approach gives slightly better results than standard approach based on projection profiles. Text line segmentation which is used in the standard projection profiles based approach is slightly slower and character segmentation is slightly faster than corresponding levels in the proposed approach. The word segmentation gives similar results from the aspect of the time complexity. The RLSA based approach used here uses two-dimensional image analysis, thus this approach gives worse results than standard projection profiles technique and the proposed

approach. It should be mentioned that provided results for processing time are obtained using the optimized implementations.

4 Conclusion

In this paper a general approach for character segmentation of machine-typed and machine-printed documents is presented. The proposed technique is semi-automatic and exploits the image processing methods in spatial domain. In section 2 complete approach is presented in details. In section 3 experimental results for time complexity and segmentation accuracy using the historical ground-truth machine-printed documents are provided. Proposed technique proved to be efficient and accurate compared with state-of-the-art methods. Since the proposed approach is designed as a part of a real-time OCR system for the needs of the "Nikola Tesla Museum", the first evaluation of the approach performances will be performed at the "Nikola Tesla Museum" in Belgrade. Our future work will be focused on optimization of the character segmentation approach including better automatic procedures (removing manual part) and its integration into the complete real-time OCR system.

Acknowledgment

This paper is supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia (Project III44006-10), Mathematical Institute of Serbian Academy of Science and Arts (SANU), The "Nikola Tesla Museum" (providing original typewritten documents of Nikola Tesla), and Pattern Recognition & Image Analysis Research Lab (PRimA) (providing ground-truth historical machine-printed documents).

References

[1] N. Bourbakis, N. Pereira, S. Mertoguno, "Hardware design of a letter-driven OCR and document processing system," Journal of Network and Computer Applications, vol. 19, no. 3, pp. 275-294, 1996.

[2] J. Mao, K. M. Mohiuddin, "Improving OCR performance using character degradation models and boosting algorithm," Pattern Recognition Letters, vol. 18, no. 11-13, pp. 1415-1419, 1997.

[3] G. Vamvakas, B. Gatos, N. Stamatopoulos, S. Perantonis, "A Complete Optical Character Recognition Methodology for Historical Documents," IAPR International Workshop on Document Analysis Systems, vol. 1, pp. 525-532, 2008.

[4] H. Fujisawa, "Forty years of research in character and document recognition-an industrial perspective," Pattern Recognition, vol. 41, no. 8, pp. 2435-2446, 2008.

[5] Y. Lu, "Machine Printed Character Segmentation - An Overview," Pattern Recognition, vol. 28, no. 1, pp. 67-80, 1995.

[6] Y. Lu, M. Shridhar, "Character segmentation in handwritten words - An overview," Pattern Recognition, vol. 29, no. 1, pp. 77-96, 1996.

[7] Á. González, L. M. Bergasa, "A text reading algorithm for natural images," Image and Vision Computing, vol. 31, no. 3, pp. 255-274, 2013.

[8] J. Lim, J. Park, G. G. Medioni, "Text segmentation in color images using tensor voting," Image and Vision Computing, vol. 25, no. 5, pp. 671-685, 2007.

[9] J. Min-Chul, S. Yong-Chul, S. N. Srihari, "Machine Printed Character Segmentation Method Using Side Profiles," Proceedings of IEEE SMC '99 Conference on Systems, Man and Cybernetics, 1999.

[10] N. Nikolaou, M. Makridis, B. Gatos, N. Stamatopoulos, N. Papamarkos, "Segmentation of historical machine-printed documents using Adaptive Run Length Smoothing and skeleton segmentation paths," Image and Vision Computing, vol. 28, no. 4, pp. 590-604, 2010.

[11] H. C. Park, S. Y. Ok, Y. J. Yu, H. G. Cho, "A word extraction algorithm for machine-printed documents using a 3D neighborhood graph model," International Journal on Document Analysis and Recognition, vol. 4, no. 2, pp. 115-130, 2001.

[12] E. B. Lacerda, C. A. B. Mello, "Segmentation of connected handwritten digits using Self-Organizing Maps," Expert Systems with Applications, vol. 40, no. 15, pp. 5867-5877, 2013.

[13] N. Stamatopoulos, B. Gatos, G. Louloudis, U. Pal, A. Alaei, "ICDAR 2013 Handwriting Segmentation Contest," 2013 12th International Conference on Document Analysis and Recognition (ICDAR), 2013.

[14] O. Surinta, M. F. Karaaba, L. R. B. Schomaker, M. A. Wiering, "Recognition of handwritten characters using local gradient feature descriptors," Engineering Applications of Artificial Intelligence, vol. 45, pp. 405-414, 2015.

[15] M. Younes, Y. Abdellah, "Segmentation of Arabic Handwritten Text to Lines," Procedia Computer Science, International Conference on Advanced Wireless Information and Communication Technologies (AWICT 2015), vol. 73, pp. 115-121, 2015.

[16] A. Antonacopoulos, D. Karatzas, "Semantics-based content extraction in typewritten historical documents," 8th International Conference on Document Analysis and Recognition (ICDAR '05), pp. 48-53, 2005.

[17] I. Bar-Yosef, A. Mokeichev, K. Kedem, I. Dinstein, U. Ehrlich, "Adaptive shape prior for recognition and variational segmentation of degraded historical characters," Pattern Recognition, vol. 42, no. 12, pp. 3348-3354, 2009.

[18] L. A. F. Fernandes, M. M. Oliveira, "Real-time line detection through an improved Hough transform voting scheme," Pattern Recognition, vol. 41, no. 1, pp. 299-314, 2008.

[19] C. Singh, N. Bhatia, A. Kaur, "Hough transform based fast skew detection and accurate skew correction methods," Pattern Recognition, vol. 41, no. 12, pp. 3528-3546, 2008.

[20] G. Bessho, K. Ejiri, J. F. Cullen, "Fast and accurate skew detection algorithm for a text document or a document with straight lines," Proc. SPIE, vol. 2181, pp. 133-140, 1994.

[21] Y. Cao, S. Wang, H. Li, "Skew detection and correction in document images based on straight-line fitting," Pattern Recognition Letters, vol. 24, no. 12, pp. 1871-1879, 2003.

[22] M. R. Gupta, N. P. Jacobson, E. K. Garcia, "OCR binarization and image pre-processing for searching historical documents," Pattern Recognition, vol. 40, no. 2, pp. 389-397, 2007.

[23] A. Sedighi, M. Vafadust, "A new and robust method for character segmentation and recognition in license plate images," Expert Systems with Applications, vol. 38, no. 11, pp. 13497-13504, 2011.

[24] M. Grafmüller, J. Beyerer, "Performance improvement of character recognition in industrial applications using prior knowledge for more reliable segmentation," Expert Systems with Applications, vol. 40, no. 17, pp. 6955-6963, 2013.

[25] N. B. Venkateswarlu, R. D. Boyle, "New segmentation techniques for document image analysis," Image and Vision Computing, vol. 13, no. 7, pp. 573-583, 1995.

[26] J. Li, M. Li, J. Pan, S. Chu, J. F. Roddick, "Gabor-based kernel self-optimization Fisher discriminant for optical character segmentation from text-image-mixed document," Optik - International Journal for Light and Electron Optics, vol. 126, no. 21, pp. 3119-3124, 2015.

[27] J. H. Bae, K. C. Jung, J. W. Kim, H. J. Kim, "Segmentation of touching characters using an MLP," Pattern Recognition Letters, vol. 19, no. 8, pp. 701-709, 1998.

[28] O. Starostenko, C. Cruz-Perez, F. Uceda-Ponga, V. Alarcon-Aquino, "Breaking text-based CAPTCHAs with variable word and character orientation," Pattern Recognition, vol. 48, no. 4, pp. 1101-1112, 2015.

[29] V. Vuckovic, B. Arizanovic, "Efficient character segmentation approach for machine-typed documents," Expert Systems with Applications, In Press, 2017 [doi: 10.1016/j.eswa.2017.03.027].

[30] W. K. Pratt, Digital Image Processing: PIKS Scientific Inside, 4th ed. New York: John Wiley & Sons, 2006.

[31] J. C. Russ, The Image Processing Handbook, 5th ed. Florida: CRC Press, 2009.

[32] R. C. Gonzalez, R. E. Woods, Digital Image Processing, 3rd ed. New Jersey: Prentice-Hall, 2008.

[33] I. T. Phillips, A. K. Chhabra, "Empirical Performance Evaluation of Graphics Recognition Systems," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 9, pp. 849-870, 1999.

[34] ABBYY FineReader OCR. <http://finereader.abbyy.com/>

[35] The OCROpus open source document analysis and OCR system. <http://code.google.com/p/ocropus>