Optimization of the Gaussian Kernel Extended by Binary Morphology for Text Line Segmentation

Darko BRODIĆ [1], Zoran MILIVOJEVIĆ [2]

[1] Technical Faculty Bor, University of Belgrade, V.J. 12, 19210 Bor, Serbia
[2] Technical College Niš, Aleksandra Medvedeva 20, 18000 Niš, Serbia

dbrodic@tf.bor.ac.rs, zoran.milivojevic@jotel.co.rs

Abstract. *In this paper, an approach for text line segmentation by algorithm with the implementation of the Gaussian kernel is presented. As a result of algorithm, the growing area around text is exploited for text line segmentation. To improve text line segmentation process, isotropic Gaussian kernel is extended by dilatation. Furthermore, algorithms with isotropic and extended Gaussian kernels are examined and evaluated under different text samples. Results are given and comparative analysis is made for these algorithms. From the obtained results, optimization of the parameters defining extended Gaussian kernel dimension is proposed. The presented algorithm with the extended Gaussian kernel showed robustness for different types of text samples.*

Keywords

OCR, document image processing, text line segmentation, Gaussian kernel, morphological operation.

1. Introduction

Text line segmentation is the major step in a document analytic procedure. It is prerequisite for high-quality optical character recognition (OCR) methods. There are a few successful techniques for printed text line segmentation. But, processing of handwritten documents has been remained a key problem in OCR [1], [2]. Most text line segmentation methods are based on the assumptions that the distance between neighboring text lines is sufficiently large as well as that text lines are reasonably straight. However, these assumptions are not always valid for handwritten documents. Hence, text line segmentation is a leading challenge in OCR.

Related work on text line segmentation can be categorized in few directions [3]: projection based methods, Hough transform methods, smearing methods, grouping methods, methods for processing overlapping and touching components, stochastic methods, and other methods.

Projection based methods have been primarily used for printed document segmentation, but it can be adapted for handwritten documents as well. They use the vertical projection profile (VPP), which is obtained by summing pixel values along the horizontal axis for each y value. This is accomplished by finding its maximum and minimum value [4]. Because of method drawbacks, short lines will provide low peaks, and very narrow lines. Hence, the method failed to be efficient for multi-skewed text lines. The Hough transform [5] is a widespread technique for finding straight lines in the images. Consequently, an image is transformed in the Hough domain. Potential alignments are hypothesized in the Hough domain and validated in the image domain. The direction for the maximum variation is determined by a cost function. The “voting” function in the Hough domain determines the slope of the straight line [6]. In smearing methods the consecutive black pixels along the horizontal direction are smeared [7]. This way, an enlarged area of black pixels is formed. It is so-called boundary growing area. Consequently, the white space between black pixels is filled with black pixels. It is valid only if their distance is within a predefined threshold limits. Grouping methods are based on building alignments by aggregating them into entities [8]. The units may be pixels or connected components, blocks or other features 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. If the nearest neighbor element belongs to another line, then the nearest neighbor joining scheme will fail to group complex handwritten units. The method for overlapping and touching components detects such components during the grouping process. At present, the conflict occurs between two alignments [9]. Further, it applies a set of rules to label overlapping or touching components. The rules use as features the density of black pixels of the component in each alignment region, alignment proximity and positions of both alignments around the component. The frontier segment position is decided by analyzing the component VPP. If VPP includes two peaks, the cut will be done in the middle of them. Otherwise, the component will be cut into two equal parts. The stochastic method is based on the probabilistic algorithm which accomplished non-linear paths between overlapping text lines. These lines are extracted through Hidden Markov Modeling (HMM) [10]. This way, the image is divided into little 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 the highest probability will cross the touching component at points with as less black pixels as possible. However, the method may fail in the case that the contact point contains a lot of black pixels. The algorithm proposed in [11] models text line detection as image segmentation problem by enhancing text line structure using a Gaussian kernel and adopting level set method to evolve text line boundaries. The author specified the method as robust for different languages, but rotating text by angle of 10° or more has an impact on reference line hit rate. The method is further investigated in [7].

In this paper, modification of the base Gaussian kernel enhanced by binary morphology is implemented. It is a simple and efficient method in terms of accuracy and computational effectiveness. Its primary role is to perform text segmentation and to estimate the skew angle of the document image. The proposed method is implemented and "measured" by different sample text examples and evaluated as well. At the end, based on measurement results, algorithm parameters optimization is proposed.

The organization of the paper is as follows. Section 2 includes proposed algorithm information and description. In Section 3 text experiments framework is defined. Further, in Section 4 results are examined, compared and discussed. In Section 5 conclusions are made.

2. Proposed Algorithm

The principal parts of OCR system are scanning, binarization, text segmentation, text parameter extraction, text recognition and conversion to ASCII. However, the document text processing procedure can be represented with three main stages: a) preprocessing, b) processing, and c) postprocessing, as shown in Fig. 1.

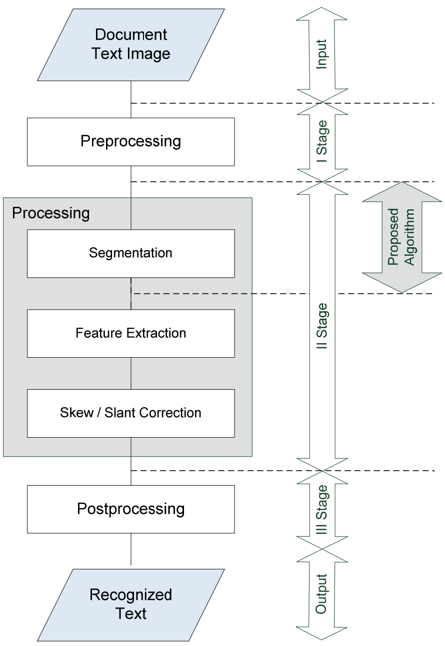

Fig. 1. Document text image processing procedure.

In the preprocessing stage, algorithms for document text image binarization and normalization are applied. During the processing stage, algorithms for text segmentation as well as for reference text line estimation and skew rate identification are enforced. After that, the skew angle is corrected. At the end, in the postprocessing stage a character recognition process is applied. The final sub-stage is the data conversion to ASCII characters.

A few assumptions should be made before an algorithm description. In this paper, there is an element of preprocessing. After that, document text image is prepared for feature extraction. The main tasks are text segmentation as well as text parameter extraction, specifically reference text line identification and skew rate estimation.

2.1 Document Text Image

At the beginning of the test process, the original image is used. The document text image is obtained as a product of the original image scanning. The document text image is a digital text image represented by matrix **D** with M rows, N columns, and intensity with L discrete levels of gray. L is the integer number from the set {0, …, 255}. Currently, $D(i,j) \in \{0, \ldots, 255\}$, where $i = 1,\ldots, M$ and $j = 1,\ldots, N$.

After applying intensity segmentation with binarization, an intensity function is converted into the binary intensity function given by:

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

where D_{th} is given by Otsu algorithm [12] or an equivalent algorithm. It represents a threshold sensitivity decision value.

Currently, the document image is represented as binary matrix **B** featuring M rows by N columns. Consequently, it consists of the only black and white pixels where value 0 represents black pixels and value 1 white pixels.

2.2 Isotropic Gaussian Kernel

Establishing distinct areas that mutually separate text lines is the primary task of the region growing algorithm for text line segmentation. Before applying the algorithm, the text consists of letters, words or groups of words. Algorithm's task is to join these text elements from the same text line into the same distinct continuous areas. Similarly, elements from the different text lines, are separated into different areas.

In this paper, the algorithm based on the analogy with Gaussian probability density function is established. This function is given by [11]:

$$g(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}} = Ae^{-\frac{x^2+y^2}{2\sigma^2}} , \tag{2}$$

where σ is the standard deviation defining a curve spread parameter and A is the amplitude of the function equal to $1/(2\pi)^{1/2}\sigma$. It is the starting point for creating an isotropic kernel. Hence, converting Gaussian function into point spread function (PSF) creates Gaussian kernel. The idea of Gaussian smoothing is to use this 2-D distribution as a PSF. Since the image is stored as a collection of discrete pixels we need to produce a discrete approximation to the Gaussian function $g(x,y)$ named $G(i,j)$ before performing the convolution. However, the Gaussian distribution is non-zero everywhere, which would require an infinitely large convolution kernel. In practice, it is effectively zero for more than about 3σ from the mean. This value represents Gaussian threshold sensitivity level L_{gts}. It truncates radius of 3σ in the kernel. All pixels that belong inside that radius form the same area with a level higher than L_{gts}. Hence, isotropic Gaussian kernel defined by $2K+1$ in x and y directions. It is given as:

$$I(i,j)=\begin{cases}1 \ , \ G(i,j)\geq L_{\text{gts}}\\ 0 \ , \ G(i,j)< L_{\text{gts}}\end{cases} . \tag{3}$$

Converting all these pixels into the same region forms the areas named boundary growing areas. Boundary growing areas form a control image with distinct objects that are prerequisite for the text segmentation of the document image. These black objects represent different text lines needful for text segmentation i.e. for disjoining text lines. Matrix **X** is created by convolving the isotropic Gaussian kernel **I** with the image represented by binary matrix **B** as follows [13]:

$$X(i,j)=\sum_{k=-K}^{K}\sum_{l=-K}^{K}B(i+k,j+l)I(k,l) \ , \tag{4}$$

where i is from K to $M-K$ and j is from K to $N-K$. Further, the elements of matrix **X** are obtained as follows: **IF** $X(i,j)\neq 0$ **THEN** $X(i,j)=1$. Matrix **X**, as well as its growing areas, are shown in Fig. 2.

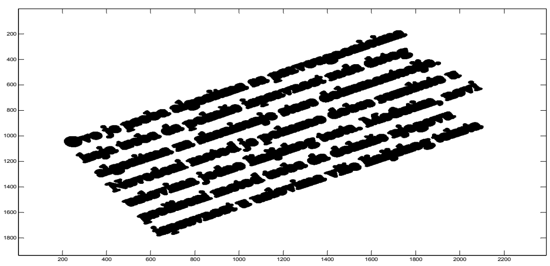

Fig. 2. Matrix **X** after applying isotropic Gaussian kernel.

2.3 Extended Gaussian Kernel

To extend growing areas an additional method is needed. Due to horizontal nature of the handwriting, the Gaussian kernel extension in the horizontal direction is preferred. Hence, the proposed kernel extension is made by convolving the isotropic Gaussian kernel with a dilatation element representing line. This line has height of 1 pixel and width of $w = 2(R-K) + 1$. Also, parameter $\lambda = R/K$ is introduced. Consequently, the extended Gaussian kernel has dimension $(2K+1) \times (2R+1)$. The difference between two kernels could be illustrated in Fig. 3.

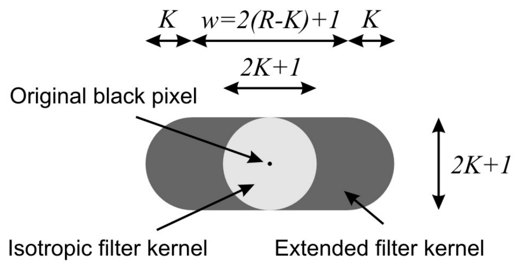

Fig. 3. Extended vs. isotropic Gaussian kernel.

The main difference between the original algorithm [11] and this approach is in the text segmentation domain. Currently, matrix **Y** is defined by convolving the extended Gaussian kernel **E** with the matrix **B** as follows [13]:

$$Y(i,j)=\sum_{k=-K}^{K}\sum_{l=-R}^{R}B(i+k,j+l)E(k,l) \ , \tag{5}$$

where i is from K to $M-K$ and j is from R to $N-R$. Further, the elements of matrix **Y** are obtained as follows: **IF** $Y(i,j)\neq 0$ **THEN** $Y(i,j)=1$. Matrix **Y** is shown in Fig. 4.

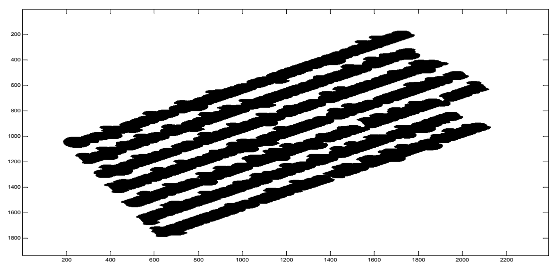

Fig. 4. Matrix **Y** after applying extended Gaussian kernel.

3. Experiments

An evaluation test framework for the text parameters extraction algorithm consists of few text experiments. It is based on the following activities: a) multi-line text segmentation test, b) multi-line waved text segmentation test, and c) multi-line fractured text segmentation test. Although, test experiments are quite diverse, theirs results are inter-related. Due to, some decision making is required at the end of the procedure. Hence, those results are combined and conclusions are made.

3.1 Multi-line Text Segmentation

Algorithm quality examination consists of a few text experiments representing test procedure. In the first group of the experiments text line segmentation quality is examined. For this purpose, as the first stage, multi-line text is used. This test is inevitable in algorithm segmentation quality assessment. Further, this experiment is significant because it is prerequisite for obtaining other text parameters. If segmentation experiment miscarry, then further process examination will be meaningless. Hence, its importance is critical. The sample multi-line text with its skew angle parameter α is shown in Fig. 5.

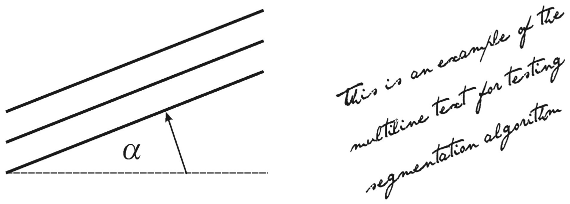

Fig. 5. Multi-line text definition and example.

The number of existing text objects in the multi-line text image relates to text segmentation quality success. Hence, the less objects the better segmentation process, except the number may not be less than the text lines number. As a quality measure, the root mean square error $RMSE_{\text{seg}}$ has been used. It is calculated as [14], [15]:

$$RMSE_{\text{seg}} = \sqrt{\frac{1}{P}\sum_{k=1}^{P}(Oref_k - Oest_k)^2} \quad , \qquad (6)$$

where k is the number of examined text samples, $Oref_k$ is the number of referent objects in the text, i.e. the number of text lines, and $Oest_k$ is the number of the obtained objects in the text by the applied algorithm.

3.2 Multi-line Waved Text Segmentation

The second text line segmentation experiment is multi-line waved text one. The sample text is formed as a group of text lines using the waved referent line for its basis. The referent line is defined by the parameter $\varepsilon = h / l$. Typically, ε is used from the set $\{1/8, 1/6, 1/4, 1/3,\ldots\}$. The sample multi-line waved text for the experiment is shown in Fig. 6.

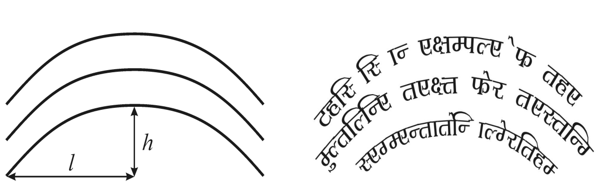

Fig. 6. Multi-line waved text definition and example.

Similarly, as a previous segmentation test, the number of existing text objects after the applied algorithm relates to the text segmentation quality success. Again, for the quality measure, the root mean square error $RMSE_{\text{seg\_wav}}$ has been used. It is calculated as [14], [15]:

$$RMSE_{\text{seg\_wav}} = \sqrt{\frac{1}{R}\sum_{l=1}^{R}(Oref_l - Oest_l)^2} \quad , \qquad (7)$$

where l is the number of the examined text samples, $Oref_l$ is the number of the referent objects in the text, i.e. the number of the text lines, and $Oest_l$ is the number of the obtained objects in the text by the applied algorithm.

3.3 Multi-line Fractured Text Segmentation

The last experiment in the first test group is the multi-line fractured text segmentation experiment. The sample text for this experiment is formed by using a referent fractured line as a basis. The fractured text referent line is defined by the slope angle ϕ, as a parameter. Typically, ϕ is used from the set $\{5°,10°,15°,20°\}$. The sample multi-line fractured text for the last segmentation experiment is shown in Fig. 7.

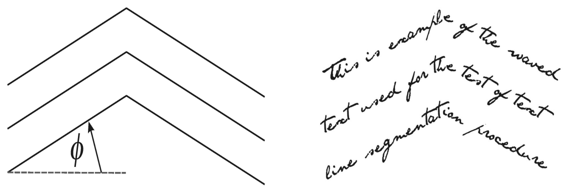

Fig. 7. Multi-line fractured text definition and example.

Again, the number of existing text objects relates to the text segmentation quality success. The root mean square error $RMSE_{\text{seg\_frac}}$ has been used. It is calculated as [14], [15]:

$$RMSE_{\text{seg\_frac}} = \sqrt{\frac{1}{Q}\sum_{m=1}^{Q}(Oref_m - Oest_m)^2} \quad , \qquad (8)$$

where m is the number of the examined text samples, $Oref_m$ is the number of the referent objects in the text, i.e. the number of the text lines, and $Oest_m$ is the number of the obtained objects in the text by the applied algorithm.

4. Results and Comparative Analysis

In all experiments, character height $H_{\text{ch}} \approx 100$ px is used. It is obtained from bounding box method [16]. Hence, K value may not exceed 1/5 of H_{ch}. In fact, a bigger K can lead to text lines merging. Multi-line text segmentation results are shown in Tab. 1. Data representing number of objects for $\alpha = 20°$ and $\alpha = 40°$ is shown in Fig. 8-9.

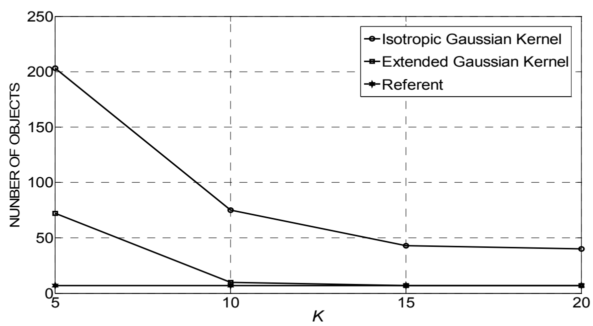

Fig. 8. Results for multi-line text experiment ($\alpha = 20°$, $\lambda = 3$).

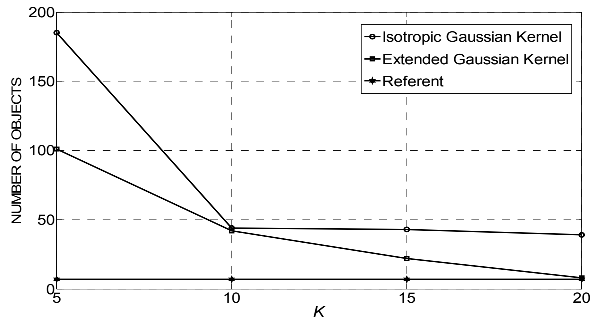

Fig. 9. Results for multi-line text experiment ($\alpha = 40°$, $\lambda = 2$).

It is obvious that obtained results for the extended algorithm are better than for the original one. Hence, the algorithm extension gives significant segmentation benefit. Still, K approaching to 20 is desired. Consequently, K should be up to 20% H_{ch} which is close to boundary condition [16]. Hence, enlarging K above 20 is “forbidden”. From all above, parameter K = {15, 20} is articulated. RMS_{seg} inspection is shown in Fig. 10-11.

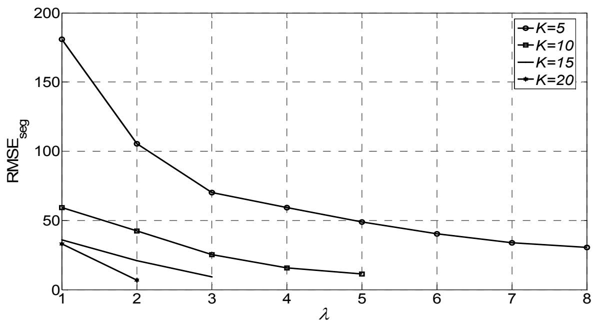

Fig. 10. $RMSE_{seg}$ vs. λ for multi-line text segmentation test.

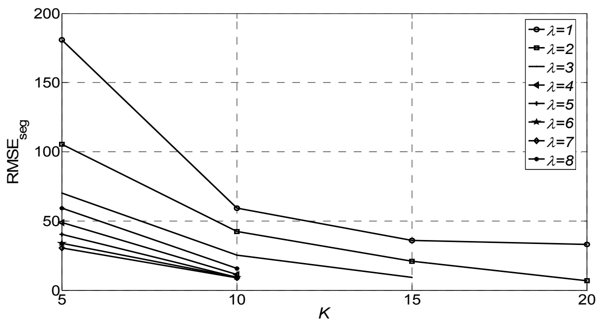

Fig. 11. $RMSE_{seg}$ vs. K for multi-line text segmentation test.

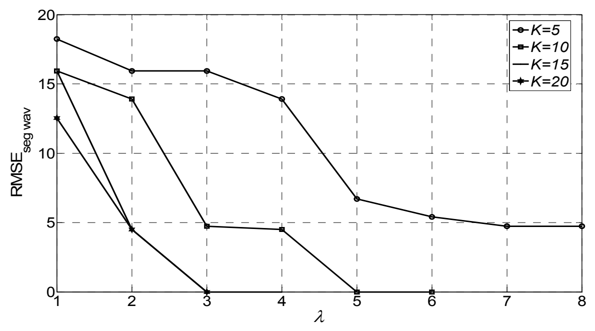

Fig. 12. $RMSE_{seg\_wav}$ vs. λ for multi-line waved text test.

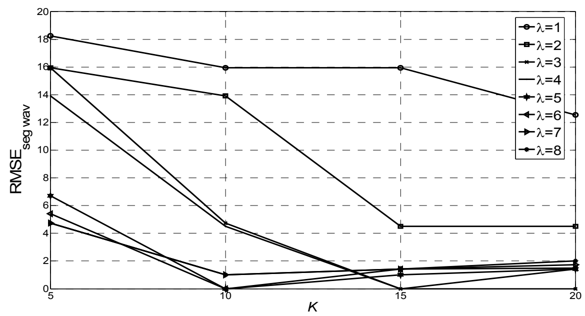

Fig. 13. $RMSE_{seg\_wav}$ vs. K for multi-line waved text test.

Further, RMS_{seg_wav} obtained from the multi-line waved text segmentation test is shown in Fig. 12-13. In this experiment, using K of 15 or 20 is the proper choice. Hence, this leads to λ of at least 3. It can be noticed that bigger λ will merge different text lines.

In the last text segmentation experiment, RMS_{seg_wav} is evaluated. It is shown in Fig. 14-15.

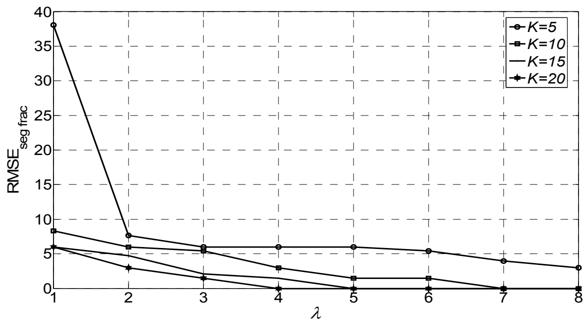

Fig. 14. $RMSE_{seg\_frac}$ vs. λ for multi-line fractured text test.

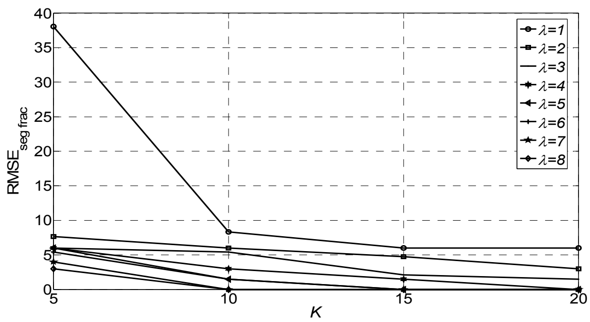

Fig. 15. $RMSE_{seg\_frac}$ vs. K for multi-line fractured text test.

Again, K of 15 or 20 is prerequisite. Still, using λ of 3 or bigger is the best choice.

At the end, the summary results for the original Gaussian kernel algorithm are not so convincing due to its faulty results obtained from the segmentation experiment. All three experiments proved strong inter-relation between by theirs obtained results. Hence, parameter value of K = {15, 20} is mandatory for the efficient text line segmentation process. Altogether, it seems that the length of the extended Gaussian kernel needs to be from 40 to 60 px or approximately from 40% to 60% H_{ch}, while the height of it should be 15% to 20% of H_{ch}. Furthermore, parameter λ completely depends on the K value. For $K = 15$, the best choice for λ is from 3 to 4, while for $K = 20$ follows λ from 2 to 3 is necessary. Obviously, smaller K such as 15 needs bigger λ, and bigger K such as 20 needs smaller λ.

From all the presented results, the algorithm based on the extended Gaussian kernel is promising. Still, a careful decision-making procedure is needed for choosing the proper values for K and λ parameters for the best segmentation results.

5. Conclusions

In this paper, an approach to the Gaussian kernel algorithm for text line segmentation is presented. The proposed improvement assumes creation of a boundary growing area around the text based on the Gaussian kernel algorithm extended by dilatation. Those growing areas form a control image with distinct objects that are prerequisite for the text line segmentation. Algorithm quality and robustness is examined by three different multi-line text experiments [14]. The results are evaluated by *RMSE* method. All obtained results are presented as well as compared with the basic Gaussian kernel method. Further, comparative analysis and discussion is made. The strength of this approach in text segmentation domain is mandatory. Its improvement is based on the expansion of the growing areas around the text. Still, a careful decision making about choosing adequate parameter values is necessary. Optimized Gaussian kernel parameters values are K, λ pairs $\{15, 3\}$, $\{15, 4\}$, $\{20, 2\}$ and $\{20, 3\}$. They are strongly related to the text letter heights. Hence, the appropriate Gaussian kernel size value is: for height 15-20% H_{ch} and for length 40-60% H_{ch}. At the and, to solve some "weakness" of the proposed algorithm, further investigation should be made toward creating optimal, adjusted and dilated Gaussian kernel rotated by the initial skew step. This way, the text line segmentation will be more errorness prone, further leading to more robust algorithm.

Appendix

Tables referencing obtained results are given below.

Number of objects								
	Isotropic	Extended Gaussian Kernel						
$\lambda = R/K$	1	2	3	4	5	6	7	8
α	$K = 5$							
0°	182	88	65	62	39	27	25	23
5°	180	82	66	62	43	29	25	23
10°	193	86	67	63	55	30	27	26
20°	203	86	72	62	55	14	12	8
30°	180	92	79	63	44	30	13	11
40°	185	101	80	60	44	41	25	13
50°	187	111	53	50	49	46	35	31
60°	182	128	66	51	51	50	48	43
70°	197	142	96	75	62	56	52	46
80°	190	170	109	99	94	93	81	77
α	$K = 10$							
0°	77	62	27	23	24	25	24	22
5°	82	62	27	19	16	9	9	9
10°	78	62	17	10	10	10	10	10
20°	75	42	10	7	7	8	9	9
30°	45	43	20	11	7	~~6~~	~~4~~	~~2~~
40°	44	42	32	11	7	5	2	2
50°	44	42	38	24	11	11	11	11
60°	45	43	42	29	29	29	29	29
70°	73	43	41	31	27	~~6~~	~~4~~	~~3~~
80°	77	45	43	33	10	~~5~~	~~3~~	~~4~~
α	$K = 15$							
0°	43	7	7	7	9	9	9	9
5°	43	7	7	8	9	9	9	9
10°	43	8	8	8	8	8	8	8
20°	43	8	7	7	9	~~6~~	~~6~~	~~6~~
30°	43	15	7	6	~~1~~	~~1~~	~~1~~	~~1~~
40°	43	22	7	~~3~~	~~2~~	~~1~~	~~1~~	~~1~~
50°	43	34	7	~~6~~	~~6~~	~~6~~	~~6~~	~~6~~
60°	43	39	16	16	16	16	16	16
70°	43	41	25	~~3~~	~~2~~	~~2~~	~~1~~	~~1~~
80°	43	41	28	~~5~~	~~4~~	~~3~~	~~3~~	~~3~~
α	$K = 20$							
0°	41	7	9	9	9	9	9	9
5°	39	7	8	9	9	9	9	9
10°	41	7	7	7	7	7	7	7
20°	40	7	7	~~6~~	~~3~~	~~3~~	~~3~~	~~3~~
30°	38	7	~~2~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~
40°	39	8	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~
50°	39	7	~~5~~	~~5~~	~~5~~	~~5~~	~~5~~	~~5~~
60°	41	9	9	9	9	9	9	9
70°	41	19	~~2~~	~~2~~	~~2~~	~~2~~	~~2~~	~~2~~
80°	41	25	~~2~~	~~2~~	~~2~~	~~2~~	~~2~~	~~2~~

Tab. 1. The results from multi-line text segmentation test (a cell with sign / represents segmentation error).

Number of objects								
	Isotropic	Extended Gaussian Kernel						
$\lambda = R/K$	1	2	3	4	5	6	7	8
ε	$K = 5$							
1/8	18	15	15	15	12	12	12	12
1/4	18	18	18	18	6	3	3	3
1/3	24	21	21	12	6	3	3	3
1/2	24	21	21	21	12	9	6	6
ε	$K = 10$							
1/8	15	15	12	12	3	3	3	3
1/4	18	18	3	3	3	3	3	3
1/3	21	12	3	3	3	3	3	3
1/2	21	21	6	3	3	3	~~1~~	~~1~~
ε	$K = 15$							
1/8	15	12	3	3	3	3	3	3
1/4	18	3	3	3	3	~~1~~	~~1~~	~~1~~
1/3	21	3	3	3	3	3	3	3
1/2	21	3	3	3	~~1~~	~~1~~	~~1~~	~~1~~
ε	$K = 20$							
1/8	12	12	3	3	3	3	3	1
1/4	18	3	3	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~
1/3	3	3	3	3	3	2	1	1
1/2	21	3	3	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~

Tab. 2. The results from the multi-line waved text experiment (a cell with sign / represents segmentation error).

Number of objects								
	Isotropic	Extended Gaussian Kernel						
$\lambda = R/K$	1	2	3	4	5	6	7	8
ϕ	$K = 5$							
5°	45	12	9	9	9	6	6	6
10°	39	9	9	9	9	9	6	6

15°	40	9	9	9	9	9	6	6
20°	40	12	9	9	9	9	9	6
ϕ	$K = 10$							
5°	12	9	6	6	6	6	3	3
10°	9	9	9	6	3	3	3	3
15°	12	9	9	6	3	3	3	3
20°	12	9	9	6	3	3	3	3
ϕ	$K = 15$							
5°	9	6	6	6	3	3	3	3
10°	9	6	3	3	3	3	3	3
15°	9	9	6	3	3	3	3	3
20°	9	9	3	3	3	3	3	3
ϕ	$K = 20$							
5°	9	6	6	3	3	3	3	3
10°	9	6	3	3	3	3	3	3
15°	9	6	3	3	3	3	3	3
20°	9	6	3	3	3	3	3	3

Tab. 3. The results from the multi-line fractured text experiment.

References

[1] AMIN, A., WU, S. Robust skew detection in mixed text/graphics documents. In *Proceedings of the 8th ICDAR '05*. Seoul (Korea), 2005, p. 247 - 251.

[2] BASU, S., CHAUDHURI, C., KUNDU, M., NASIPURI, M., BASU, D. K. Text line extraction from multi-skewed handwritten document. *Pattern Recognition*, 2006, vol.40, p. 1825-1839.

[3] LIKFORMAN-SULEM, L., ZAHOUR, A., TACONET, B. Text line segmentation of historical documents: A Survey. *International Journal on Document Analysis and Recognition*, 2007, vol. 9, no. 2-4, p. 123-138.

[4] SILVA, L. F., CONCI, A., SANCHEZ, A. Automatic discrimination between printed and handwritten text in documents. In *Proceedings of 22nd BSCGIP*. Rio de Janeiro (Brazil), 2009, p. 261-267.

[5] BALLARD, D. H. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 1981, vol. 13, no. 2, p. 111-122.

[6] AMIN, A., FISCHER, A. A. Document skew detection method using the Hough transform. *Pattern Analysis & Applications*, 2000, vol. 3, no. 3, p. 243-253.

[7] SHI, Z., GOVINDARAJU, V. Line separation for complex document images using fuzzy runlength. In *Proceedings of the IWDIAL '04*. Palo Alto (USA), 2004, p. 306-312.

[8] 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), p. 21-38. Paris: Europia, 1994.

[9] ZAHOUR, A., TACONET, B., MERCY, P., RAMDANE, S. Arabic handwritten text-line extraction. In *Proceedings of the 6th ICDAR '01*. Seattle (USA), 2001, p. 281-285.

[10] KOSHINAKA T., KEN'ICHI I., AKITOSHI O. An HMM-based Text Segmentation Method using Variational Bayes Approach. *IEIC Technical Report* 2004, pp.19-24, Vol.104, No.87, pp.19-24.

[11] LI, Y., ZHENG, Y., DOERMANN, D., JAEGER, S. A new algorithm for detecting text line in handwritten documents. In *Proceedings of the 18th ICPR '06*. Hong Kong (China), 2006, vol. 2, p. 1030-1033.

[12] OTSU, N. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 1979, vol. 9, no. 1, p. 62-66.

[13] GONZALES, R. C., WOODS, R. E. *Digital Signal Processing*, 2nd Ed. Upper Saddle River: Prentice-Hall, 2002, p. 116-118.

[14] BRODIĆ, D., MILIVOJEVIĆ, D. R., MILIVOJEVIĆ, Z. Basic test framework for the evaluation of text line segmentation and text parameter extraction. *Sensors*, 2010, vol. 10, no. 5, p. 5263-5279.

[15] BOLSTAD, W. M. *Introduction to Bayesian Statistics*. New York: John Wiley & Sons, 2004, p. 42-44.

[16] 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, p. 1-6.

About Authors

Darko BRODIĆ was born in Konjic, BiH, in 1963. He received his B.E., and M.E. from the Faculty of Electrical Engineering, University of Sarajevo in 1987 and 1990, respectively. Currently, he is with the University of Belgrade, Technical Faculty Bor. He is working toward Ph.D. thesis. His research interests include document image processing. He is the author and coauthor of over 30 journal and conference papers as well as many books.

Zoran MILIVOJEVIĆ was born in Svrljig, Serbia, in 1959. He received his B.E., M.E., and Ph.D degrees in Electrical Engineering from the Faculty of Electronic Engineering, University of Niš in 1984, 1994 and 2002, respectively. He is a Professor at Technical College in Niš. His primary research interests are digital signal processing: algorithms and applications. He is the author and coauthor of over 130 journal and conference papers.