

Approach to the Improvement of the Text Line Segmentation by Oriented Anisotropic Gaussian Kernel

D. Brodic

*University of Belgrade, Technical Faculty Bor,
V.J.12, 19210 Bor, Serbia, phone: +381 30 424 555, e-mail: dbrodic@tf.bor.ac.rs*

Z. N. Milivojevic

*Technical College Nis,
Aleksandra Medvedeva 20, 18000 Nis, Serbia, phone: +381 18 588 211, e-mail: zoran.milivojevic@jotel.co.rs*

D. R. Milivojevic

*Mining and Metallurgy Institute, Department of Informatics,
Zeleni bulevar bb, 19210 Bor, Serbia, phone: +381 30 435 109, e-mail: dragan.milivojevic@irmbor.co.rs*

crossref <http://dx.doi.org/10.5755/j01.eee.118.2.1181>

Introduction

Text line segmentation is a major step in a document analytic procedure. It is prerequisite for the valid optical character recognition (OCR) process. In addition, the text line segmentation and character recognition are dependent tasks as well [1].

There are a few successful techniques for printed text line segmentation. However, processing of handwritten documents has been remained a key problem in OCR [2, 3]. Most text line segmentation methods are based on the assumptions that distance between neighboring text lines is sufficiently large and 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 [1]: projection based methods, Hough transform methods, smearing methods, grouping methods, methods for processing overlapping and touching components, stochastic methods, and others.

Conventionally, text is written around the horizontal axis. Smearing methods exploited this text property. Hence, they smeared consecutive black pixels representing text along the horizontal direction. If the distance between the white space is within predefined threshold, it is filled with black pixels. The bounding boxes of the connected components in the smeared image which represents control image are considered as text lines.

Algorithm proposed in [4] model 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. Firstly, it converts a binary image to gray scale using Gaussian window, which enhances text line structures. Further, text line is extracted by evolving an initial estimate using the level set method. Specified method proved to be robust for different languages under reasonable variation of the skew angles, character sizes, and noise. Still, rotating text by angle of 10° or more has an impact on reference line hit rate. Similar approach for text image segmentation is partly exploited in [5]. Some theoretical extensions are given in the form of Gabor transformation [6]. Further, original method is intensively investigated in [7].

In this paper, the approach to the oriented anisotropic Gaussian kernel is proposed. Firstly, preprocessing is made by creating bounding boxes over each text object. According to this, its text object skew is evaluated by moments. These values determine the text orientation. Hence, they are inputs into an oriented Gaussian Kernel based algorithm. Furthermore, the benefits of the rotated over non-rotated Gaussian kernel for text line segmentation are also investigated. It is evaluated by printed and handwritten text samples. At the end, algorithm parameters optimization linked with Gaussian kernel are proposed.

The article is divided into different Sections: Section 2 describes algorithms i.e. algorithm based on non-rotated as well as rotated Gaussian kernel. Section 3 defines testing process and shows measurement results. Section 4 presents comparative analysis. Section 5 makes

conclusions.

Proposed algorithm

Document Text Image. At the beginning of the process, continuous image is used. Document text image is obtained as product of image scanning. Document text image is digital text image represented by matrix \mathbf{D} with M rows, N columns, and intensity with L discrete levels of gray. L is the integer number from the set $\{0, \dots, 255\}$. Currently, $D(i,j) \in \{0, \dots, 255\}$, where $i = 1, \dots, M$ and $j = 1, \dots, N$.

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

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

where D_{th} is given by Otsu algorithm [8] or equivalent algorithm [9–11]. It represents threshold sensitivity decision value.

Currently, document text image is represented as binary matrix \mathbf{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 pixel. Illustration of handwritten text binary image is shown in Fig. 1.

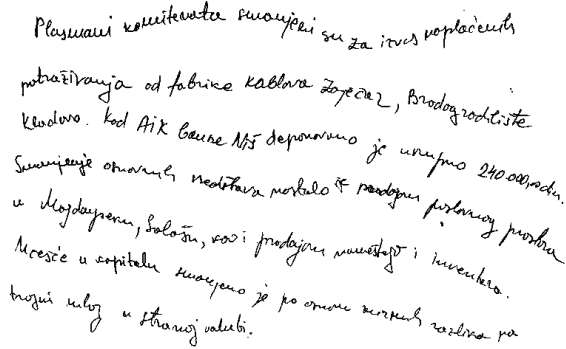


Fig. 1. Binary document image

Anisotropic Gaussian Filter. Establishing distinct regions that mutually separate text lines is the primary task of the optical character recognition (OCR) algorithm.

In this paper, algorithm based on the analogy with Gaussian probability density function (PDF) is established. The function is given by [12]

$$G(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})}, \quad (2)$$

where \mathbf{x} and $\boldsymbol{\mu}$ are column vectors and Σ is covariance matrix. For 2-D column vector \mathbf{x} is given as

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}, \quad (3)$$

while vector $\boldsymbol{\mu}$ as

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}. \quad (4)$$

Furthermore, covariance matrix Σ is given as

$$\Sigma = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}, \quad (5)$$

while its determinant $|\Sigma|$ as

$$|\Sigma| = \sigma_x^2 \sigma_y^2. \quad (6)$$

In (5) and (6) σ_x and σ_y represent the standard deviation defining curve spread parameter and μ_x and μ_y represent the mean in x and y direction, respectively. Further, if $\sigma_x = \sigma_y$ this is isotropic Gaussian PDF, elsewhere it is anisotropic. However, (2) is starting point for creating anisotropic kernel. Hence, converting Gaussian PDF into point-spread function (PSF) creates anisotropic 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 of the Gaussian function $G(\mathbf{x})$ named $G(i,j)$ prior to 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 $3\sigma_x$ and $3\sigma_y$ from the mean in x and y direction, respectively [7]. These values represent Gaussian threshold sensitivity level L_{gtsx} and L_{gtsy} . It truncates $3\sigma_x$ in x direction and $3\sigma_y$ in y direction of the kernel forming the ellipse. All pixels which are inside ellipse form the same region with level higher than L_{gtsx} or L_{gtsy} . Hence, anisotropic Gaussian kernel $G(i,j)$ is defined by $2P+1$ in x and $2R+1$ in y directions.

Currently, all these pixels are converted into the same regions thus forming boundary-growing regions. They represent control image with distinct objects. These objects are prerequisite for the text line segmentation. Furthermore, matrix \mathbf{X} is created by convolving the isotropic Gaussian kernel \mathbf{G} with the image represented by binary matrix \mathbf{B} as follows [13]

$$X(i, j) = \sum_{k=-P}^P \sum_{l=-R}^R B(i+k, j+l)G(K, l), \quad (7)$$

where $i = P, \dots, M-P$ and $j = R, \dots, N-R$. Further, elements of matrix \mathbf{X} are obtained as follows:

<pre> IF X(i, j) ≠ 0 THEN X(i, j) = 1 ELSE X(i, j) = 0 ENDIF. </pre>
--

Oriented Anisotropic Gaussian Filter. Oriented anisotropic Gaussian kernel forms extended anisotropic Gaussian kernel \mathbf{E} . Hence, proposed kernel extension is made by rotating anisotropic Gaussian kernel \mathbf{G} for the angle ϕ . Due to the nature of rotation, kernel is extended in x direction and diminished in y direction. This can be described as follows

$$\mathbf{E}(\mathbf{x}) = \mathbf{T}\mathbf{G}(\mathbf{x}), \quad (8)$$

where \mathbf{T} represents transformation matrix given by [14]

$$\mathbf{T} = \begin{bmatrix} \cos \phi & \sin \phi \\ -\sin \phi & \cos \phi \end{bmatrix}. \quad (9)$$

Angle ϕ is obtained by binary moments ϕ_M [15, 16]. Firstly, to localize text object orientation, all text objects are separated into different bounding boxes [17]. In such way, text image is split into small text objects with different attributes. Currently, for each text object the binary moment representing its orientation is calculated. It presents the evaluated orientation of each of local text's objects. Illustration of this preprocessing procedure is shown in Fig. 2.

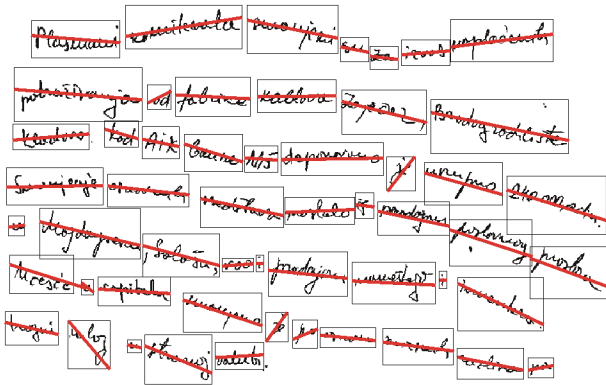


Fig. 2. Bounding boxes over each separated text object linked with its evaluated local orientation (red lines)

Furthermore, new kernel dimensions are given as $2S+1$ in x and $2T+1$ in y direction, respectively. Difference between two kernels is shown in Fig. 3.

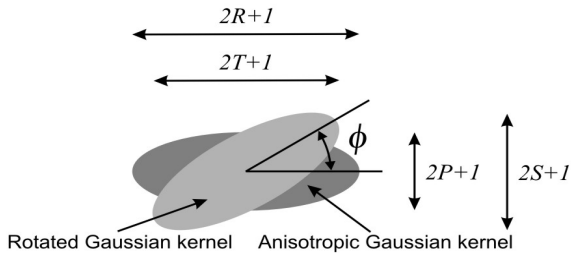


Fig. 3. Anisotropic and rotated Gaussian kernel

Main difference between original algorithm [4] and our approach is in text segmentation domain. Currently, matrix \mathbf{Y} is defined by convolving the rotated anisotropic Gaussian kernel \mathbf{E} with the matrix \mathbf{B} as follows [13]

$$Y(i, j) = \sum_{k=-S}^S \sum_{l=-T}^T B(i+k, j+l) E(K, l), \quad (10)$$

where $i = S, \dots, M-S$ and $j = T, \dots, N-T$.

Further, elements of matrix \mathbf{Y} are obtained by the binarization process as follows:

```

IF Y(i, j) ≠ 0
  THEN Y(i, j) = 1
  ELSE Y(i, j) = 0
ENDIF.

```

The illustration of above text line segmentation procedure is shown in Fig. 4.

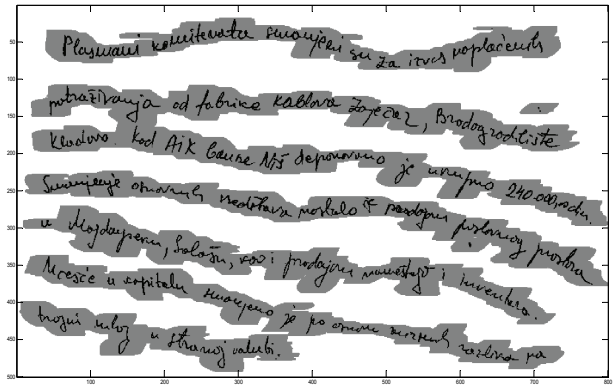


Fig. 4. Text segmentation by algorithm based on oriented anisotropic Gaussian kernel.

Experiments and Evaluation

Evaluation of the text line segmentation consists of testing samples based on IAM handwriting database [18] and extended by own test framework [19]. Testing of the algorithm represents the process of the applying algorithm to the proposed text samples. As the implication of the test, the new growing region around the text is arisen, this process leads to new text objects configuration. In an ideal circumstance the number of newly arranged objects corresponds to the correct number of text lines. To make valid algorithm evaluation following text elements should be defined [20]:

- Initial objects number O_{init} ;
- Detected objects number O_{det} ;
- Referent objects number O_{ref} .

Initial objects O_{init} represent the starting number of objects in referent sample text. It is calculated as the counted number of text objects in the starting sample text. After applying the algorithm over sample text, the number of text objects is changed. Consequently, many text objects are mutually merged by the influence of the text segmentation algorithm. Currently, the number of text objects is given as the number of detected objects O_{det} . Obviously, the number of detected objects should be lesser than or equal to the number of initial objects, i.e. $O_{det} \leq O_{init}$ should be valid. Basically, in the initial sample text there is distinct number of text lines. The task of the text segmentation algorithm is to segment text lines hitting or missing this number of lines. Hence, this number of real text lines should be represented as the target number in referent sample text. It is called referent number of objects O_{ref} . The algorithm efficiency is evaluated by comparison of the referent and detected number of objects per each text line.

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 $O_{corrindet}$. However, all others are defined as error. Segmentation errors are present in the following circumstances:

- Over-segmentation detected text lines $O_{overlindet}$ (Split lines error i.e. SLE [21]);

- Under-segmentation detected text lines $O_{underlindet}$ (Joined lines error, i.e. *JLE* [21]);
- Detected text lines with mutually inserted words from different text lines $O_{mixedlindet}$ (Lines including outlier words, i.e. *LIOW* [21]).

Split lines error represents the text lines that are wrongly divided by algorithm in two or more components, i.e. 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. All of these circumstances are illustrated in Fig. 5.

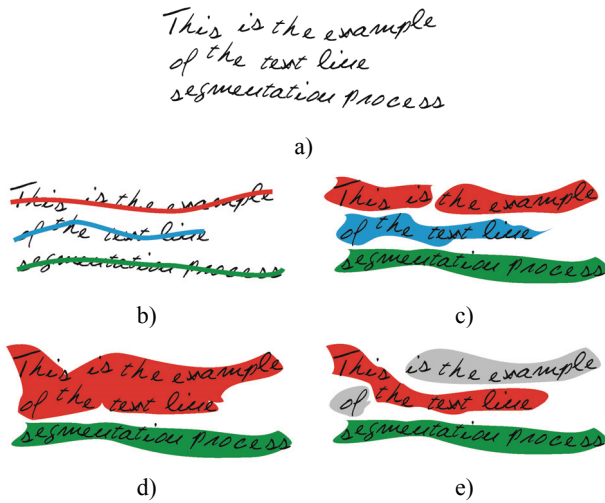


Fig. 5. Text line segmentation: (a) Original text, (b) Original text with referent objects i.e. correctly segmented text lines, (c) Over-segmentation text lines, (d) Under-segmentation text lines, and (e) Text lines with mutually inserted words from different text lines.

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 referent objects, then the algorithm is more efficient. Following elements are introduced in order to evaluate the algorithm's efficiency:

- Segmentation line hit rate, i.e. *SLHR*;
- Over-segmentation line hit rate, i.e. *OSLHR*;
- Under-segmentation line hit rate, i.e. *USLHR*;
- Mixed line hit rate, i.e. *MSLHR*;
- Segmentation root mean square error (*RMSE*), i.e. $RMSE_{seg}$ [19].

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

$$SLHR = 1 - |RE| = 1 - \left| \frac{O_{ref} - O_{corrindet}}{O_{ref}} \right|. \quad (11)$$

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

merging all objects of the text line into one.

OSLHR represents the ratio of the number of over-segmented text lines over the total number of text lines in the referent sample text. It is defined as

$$OSLHR = 1 - |RE| = 1 - \left| \frac{O_{ref} - O_{overlindet}}{O_{ref}} \right|. \quad (12)$$

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 considered 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 referent sample text. It is defined as

$$USLHR = 1 - |RE| = 1 - \left| \frac{O_{ref} - O_{underlindet}}{O_{ref}} \right|. \quad (13)$$

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

$$MSLHR = 1 - |RE| = 1 - \left| \frac{O_{ref} - O_{mixedlindet}}{O_{ref}} \right|. \quad (14)$$

At the end, the number of detected and referent text objects per each text line are compared as well. Hence, the number of referent text objects per line is equal to 1. The variance evaluation is given by $RMSE_{seg}$ [19]

$$RMSE_{seg} = \sqrt{\frac{1}{N} \sum_{i=1}^N (O_{i,ref} - O_{i,est})^2}, \quad (15)$$

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

Results and comparative analysis

In [7], optimized parameter set for the text attributes by [22] is proposed. This set is given by two parameters: P and λ , where $2P + 1$ represents x dimension of the Gaussian kernel. Consequently, λ represents the ratio of the y and x dimension of the Gaussian kernel, i.e. $\lambda = R / P$. Furthermore, it is rotated for the angle ϕ . The results for the test framework [18-19] which consist of 576 text lines with different scripts: Roman Latin, Cyrillic, Glagolitian and Bangla is given in Table 1. – Table 4.

Table 1. *SLHR* (%)

P	P_1		P_2		P_3	
λ	3	5	3	5	3	5
$\phi = 0^\circ$	81.25	95.83	95.83	72.92	81.25	58.33
$\phi = \phi_M / 2$	83.33	100.00	100.00	95.83	79.17	72.92
$\phi = \phi_M$	91.67	100.00	95.83	95.83	77.08	77.08

Table 2. *OSLHR* (%)

P	P_1		P_2		P_3		
	λ	3	5	3	5	3	5
$\phi = 0^\circ$	18.75	2.08	0.00	0.00	0.00	0.00	0.00
$\phi = \phi_M / 2$	16.67	0.00	0.00	0.00	0.00	0.00	0.00
$\phi = \phi_M$	8.33	0.00	0.00	0.00	0.00	0.00	0.00

Table 3. *USLHR* (%)

P	P_1		P_2		P_3	
	λ	3	5	3	5	3
$\phi = 0^\circ$	0.00	2.08	4.17	27.08	18.75	41.67
$\phi = \phi_M / 2$	0.00	0.00	0.00	4.17	20.83	27.08
$\phi = \phi_M$	0.00	0.00	4.17	4.17	22.92	22.92

Table 4. *RMSE_{seg}*

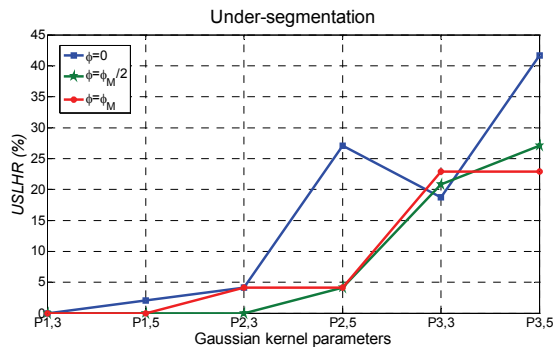
P	P_1		P_2		P_3	
	λ	3	5	3	5	3
$\phi = 0^\circ$	0.61	0.20	0.20	0.52	0.43	0.65
$\phi = \phi_M / 2$	0.54	0.00	0.00	0.20	0.46	0.52
$\phi = \phi_M$	0.29	0.00	0.20	0.20	0.48	0.48

In Table 1. – Table 4 P_1 , P_2 and P_3 ($P_1 < P_2 < P_3$) represent different values of parameter P , while kernel is rotated for the angles 0° , $\phi_M/2$ and ϕ_M . Furthermore, *SLHR* results improvement among different kernel rotation angles $\phi_M/2$ and ϕ_M compared to non-oriented kernel is given in table 5.

Table 5. *SLHR* (%) improvement compared to non-oriented kernel

P	P_1		P_2		P_3	
	λ	3	5	3	5	3
$\phi = \phi_M / 2$	+2.08	+4.17	+4.17	+22.91	-2.08	+14.59
$\phi = \phi_M$	+10.42	+4.17	+0.00	+22.91	-4.17	+18.75

In Fig. 6 and 7 *USLHR* as well as *SLHR* improvement are shown, respectively.

**Fig. 6.** *USLHR* (%) for non-oriented (blue line) and oriented Gaussian kernel (red and green line)

Under-segmentation process is unwelcomed. Hence, it should be avoided by any means. The bigger value of the parameter P triggers it. Hence, for the proper selection of the parameter careful decision-making is needed.

Consequently, decision of the P value is mandatory for the efficient of the algorithm for the text line segmentation. However, careful examination of the results presented in Fig. 7 lead to the conclusion that using oriented Gaussian kernel will reduce it. Accordingly, any

orientation of the kernel toward the referent value is welcomed.

**Fig. 7.** Application of the non-oriented (red line) and rotated Gaussian kernel (green and blue line)

Conclusions

In this paper, an approach to Gaussian kernel algorithm for text line segmentation is presented. The proposed improvement method assumes creation of boundary growing region around text based on Gaussian kernel algorithm extended by incorporating the local orientation. Those growing regions form control image with distinct objects that are prerequisite for text line segmentation. Algorithm quality and robustness is examined combined test framework. Results are evaluated by proposed method. All results are presented as well as compared with non-oriented anisotropic Gaussian kernel method. Furthermore, 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 regions under specified angle around the text. Still, careful decision-making about choosing adequate parameter values is necessary.

References

1. **Likforman–Sulem L., Zahour A., Taconet B.** Text Line Segmentation of Historical Documents: A Survey // International Journal on Document Analysis and Recognition. – Heidelberg: Springer–Verlag, 2007. – No. 2–4(9). – P. 123–138.
2. **Amin A., Wu S.** Robust Skew Detection in mixed Text/Graphics Documents // Proceedings of 8th International Conference on Document Analysis and Recognition. – Seoul, Korea, 2005. – P. 247–251.
3. **Basu S., Chaudhuri C., Kundu M., Nasipuri M., Basu D. K.** Text Line Extraction from Multi–Skewed Handwritten Documents // Pattern Recognition. – Amsterdam: Elsevier, 2007. – No. 6(40). – P. 1825–1839.
4. **Li Y., Zheng Y., Doermann D., Jaeger S.** A New Algorithm for Detecting Text Line in Handwritten Documents // Proceedings of 18th International Conference on Pattern Recognition. – Hong Kong, China, 2006. – Vol. 2. – P. 1030–1033.
5. **Laptik R., Navakasas D.** Application of Ant Colony Optimization for Image Segmentation // Electronics and Electrical Engineering. – Kaunas: Technologija, 2007. – No. 8(80). – P. 13–18.
6. **Kranauskas J.** Accelerated Calculation of Gabor Features in Spatial Domain // Electronics and Electrical Engineering. – Kaunas: Technologija, 2010. – No.1(97). – P. 39–44.

7. **Brodić D.** Optimization of the Anisotropic Gaussian Kernel for Text Segmentation and Parameter Extraction // Proceedings of Theoretical Computer Science, Calude C. S. & Sassone V., Eds. – Boston: Springer, 2010. – Vol. 323. – P. 140–152.
8. **Otsu N.** A Threshold Selection Method from Gray-level Histograms // IEEE Transactions on Systems, Man, and Cybernetics. – Piscataway: IEEE, 1979. – No. 1(9). – P. 62–66.
9. **Sauvola L., Pietikainen M.** Adaptive Document Image Binarization // Pattern Recognition. – Amsterdam: Elsevier, 2000. – No. 2(33). – P. 225–236.
10. **Bukhari S. S., Shafait F., Bruesl T. M.** Adaptive Binarization of Unconstrained Hand-Held Camera-Captured Document Images // Journal of Universal Computer Science. – Graz: TU Graz, 2008. – No. 18(15). – P. 3343–3363.
11. **Khashman A., Sekeroglu B.** Document Image Binarisation Using a Supervised Neural Network // International Journal of Neural Systems. – Singapore: World Scientific, 2008. – No. 5(18). – P. 405–418.
12. **Jähne B.** Digital Image Processing // Berlin-Heidelberg: Springer-Verlag, 2005.
13. **Gonzales R. C., Woods R. E.** Digital Signal Processing, 2nd ed. – Upper Saddle River: Prentice-Hall, 2002.
14. **Terrell G. R.** Mathematical Statistics: A Unified Introduction. – New York: Springer-Verlag, 1999.
15. **Flusser J., Suk T.** Degraded Image Analysis: An Invariant Approach // IEEE Transactions on Pattern Analysis and Machine Intelligence. – IEEE, 1998. – No. 6(20). – P. 590–603.
16. **Dey S., Bhattacharya B. B., Kundu M. K.** A Simple Architecture for Computing Moments and Orientation of an Image // Fundamenta Informaticae – Amsterdam: IOS, 2002. – No. 4(52). – P. 285–295.
17. **Brodić D., Milivojević, Z.** A New Approach to Water Flow Algorithm for Text Line Segmentation // Journal of Universal Computer Science (J.UCS). – Graz: 2011. – No. 1(17). – P. 30–47.
18. **Zimmermann M., Bunke, H.** Automatic segmentation of the IAM off-line database for handwritten English text // Proceedings of 16th International Conference on Pattern Recognition. – Quebec, Canada, 2002. – Vol. 4. – P. 35–39.
19. **Brodić D., Milivojević D. R., Milivojević Z.** Basic Test Framework for the Evaluation of Text Line Segmentation and Text Parameter Extraction // Sensors. – Basel: MDPI, 2010. – No. 5(10). – P. 5263–5279.
20. **Brodić D.** Advantages of the Extended Water Flow Algorithm for Handwritten Text Line Segmentation // Pattern Recognition and Machine Intelligence. – Berlin Heidelberg: Springer-Verlag, 2011. – Vol. 6744. – P. 418–423.
21. **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 // Proceedings of the 1st Workshops on Image Processing Theory, Tools and Applications (IPTA'2008). – Sousse, Tunisia, 2008. – P. 1–6.
22. **Brodić D., Dokić B.** Initial Skew Rate Detection Using Rectangular Hull Gravity Center // 14th International Conference on Electronics (E'2010). – Vilnius, Lithuania, 2010. – P. 1–6.

Received 2010 10 19

Accepted after revision 2011 07 27

D. Brodic, Z. N. Milivojevic, D. R. Milivojevic. Approach to the Improvement of the Text Line Segmentation by Oriented Anisotropic Gaussian Kernel // Electronics and Electrical Engineering. – Kaunas: Technologija, 2012. – No. 2(118). – P. 89–94.

The paper presents the approach to the algorithm for text line segmentation based on the anisotropic Gaussian kernel. As a result of this algorithm the growing region around text is exploited. Furthermore, anisotropic Gaussian kernel is rotated to improve text line segmentation process. Text objects orientation is evaluated by binary moments. For test purposes algorithm is evaluated under different text samples. From the obtained results comparative analysis between algorithm with anisotropic and oriented Gaussian kernel is made. At the end, benefits of the extended approach are revealed. Ill. 7, bibl. 22, tabl. 5 (in English; abstracts in English and Lithuanian).

D. Brodic, Z. N. Milivojevic, D. R. Milivojevic. Teksto parametru ekstrahavimas sukanant anizotropinį Gauso branduolį // Elektronika ir elektrotechnika. – Kaunas: Technologija, 2012. – Nr. 2(118). – P. 89–94.

Pateikiamas teksto eilutės segmentavimo algoritmas, pagrįstas anizotropiniu Gauso branduoliu. Šio algoritmo rezultatas yra didėjančios aplink tekstą srities panaudojimas. Be to, anizotropinis Gauso branduolys yra pasuktas, siekiant pagerinti teksto eilutės segmentacijos procesą. Tekstinių objektų orientacija yra įvertinama dvejetainiais momentais. Bandymų tikslais algoritmas yra vertintas naudojant skirtingus teksto pavyzdžius. Remiantis gautais rezultatais atlikta lyginamoji algoritmų su anizotropiniu ir orientuotu Gauso branduoliais analizė. Il. 7, bibl. 22, lent. 5 (anglų kalba; santraukos anglų ir lietuvių k.).