COMPARISON OF THE GAUSSIAN WINDOW AND WATER FLOW ALGORITHM FOR PRINTED AND HANDWRITTEN TEXT PARAMETERS EXTRACTION

D. Brodić[1] & Z. Milivojević[2]

[1]University of Belgrade, Technical Faculty Bor
Vojske Jugoslavije 12, 19210 Bor, Serbia, e-mail: dbrodic@tf.bor.ac.rs
[2]Technical College Niš
Aleksandra Medvedeva 20, 18000 Niš, Serbia, e-mail: zoran.milivojevic@vtsnis.edu.rs

Abstract

In this paper two algorithms: Gaussian window and water flow algorithm for text parameters identification and extraction are presented. As a result of algorithms extended text area is formed. It is used for text parameters calculation and estimation. Using numerical methods such as least squares and interpolation, reference text line and skew rate are calculated, estimated and extracted.

Both algorithms are analyzed, examined and evaluated under different printed and "handwritten" text samples. Result from experiments pointed out their strengths and weakness. Finally, both algorithm results are given, com-pared and summarized.

Algorithms showed robustness for wide variety kinds of skewness.

1. INTRODUCTION

Sample printed text is usually well formed and characterized by strong regularity in shape. Different text lines have similar orientation. Hence, they have similar or equal skewness. Due to, their orientation hasn't variability on same text page. Descendants and ascendants from neighbour text lines are sufficiently apart from. So, they aren't mix up mutually. It means text distance between lines is big enough to regularly split up text lines. Similarly, word in text lines are formed regularly with pretty similar distance and inter word spacing is decent as well.

Unlike, handwritten text is fully or partially cursive text. It tends to be differently oriented and skewed. Text lines in the handwritten documents are primarily curvilinear and close to each other. Descendants and ascendants from neighbour text lines are occasionally mix up. Text distance between lines is close to each others. So, text lines get into each other. Word in text lines aren't formed regularly and their distance is different. But, similarly to printed text, handwritten text inter word spacing is tolerable. But, appearance of different orientation skewed lines and text lines close to each other made the handwritten text to be less readable.

Previous work on text line parameters identification and extraction can be categorized in few directions:

– Histogram analysis,
– K-nearest neighbour clustering,
– Projection profile,
– Fourier transform,
– Cross-correlation,
– Other models.

In [1] is mentioned proposed technique of reference line extraction based on identifying valleys of horizontal pixel density histogram. Method failed due to multi-skewed text lines.

K-nearest neighbour clustering method i.e. docstrum [2] is by product of a larger page layout analysis system, which assumed that only text is being processed. The connected components formed by the nearest neighbours clustering are essentially characters only. The method is suitable for finding skew angle. But, it is limited to Roman languages due to poor text line segmentation.

Another method proposed in [2, 3] deal with simple multi-skewed text. It uses as a basis simple type of Hough transform for straight lines. But, it is too specific.

The Fourier transform method is a representation in the Fourier domain of the projection profile method in the pixel domain. The results are mathematically identical, but Fourier transform is only different approach to the same text and document properties projection profile is based upon [2].

The cross-correlation method calculates both horizontal and vertical projection profiles and then compares the shift inter-line cross-correlation to

determine the skew rate. Although method can handle complex layout structure documents, applied range is limited to (-10°, 10°) [2].

Algorithm proposed by [4] model text line detection as an image segmentation problem by enhancing text line structure using a Gaussian window and adopting the level set method to evolve text line boundaries. Author specified method as robust for different languages, but rotating text by an angle of 10° or more has an impact on reference line hit rate.

Method of identifying words contour area as a start of detecting baseline point proposed in [5]. But, the assumptions made on the definition of word elements are too specific.

Method [1] hypothetically assumed a flow of water in a particular direction across image frame in a way that it faces obstruction from the characters of the text lines. This method is adopted in [6]

In this paper, the base modification of methods proposed in [1] and [4] are implemented, analysed, examined and compared.

This paper is organized as follows: Section 2 includes brief description and information on proposed algorithms. In Section 3 text experiments are defined. Further, in Section 4 given results are examined, compared and discussed. In Section 5 conclusion is made as well as further investigation direction.

2. PROPOSED ALGORITHMS

Document text image identification procedure consists of three main stages as shown in Figure 1.

In preprocessing stage, algorithm for document text image binarization and normalization is applied. Now, preprocessing text is prepared for segmentation, feature extraction and character recognition. During the processing stage, algorithms for text segmentation as well as for skew and reference text line identification are enforced. After that, reference text based on skew and stroke angle, is straightened and repaired. Finally, in postprocessing stage character recognition process is applied.

In this paper, elements of processing stage are employed i.e. feature extraction. A few assumptions should be made before defining algorithm. We suppose that there is an element of preprocessing and processing. After that document text image is prepared for feature extraction. Hence, it represents distinct entity consists of group of words.

Document text image is an input of text greyscale image described by following intensity function:

$$I(l, k) \in [0,..,255], \quad (1)$$

where $l \in [0, N-1]$ and $k \in [0, M-1]$.

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

$$I_{bin}(l, k) = \begin{cases} 1 \text{ for } I(l,k) \geq I_{th} \\ 0 \text{ for } I(l, k) < I_{th} \end{cases} \quad (2)$$

where I_{th} is given by Otsu algorithm [8].

Now, separated and extracted text line is represented as digitized document image by M x N dimension matrix X. Each word in document image consists of black points i.e. pixels. Every point is represented by number of coordinate pairs such as:

$$X(i, j) \in [0,1], \quad (3)$$

where $i = 1, \ldots, N$, $j = 1, \ldots, M$ of matrix X [9, 10].

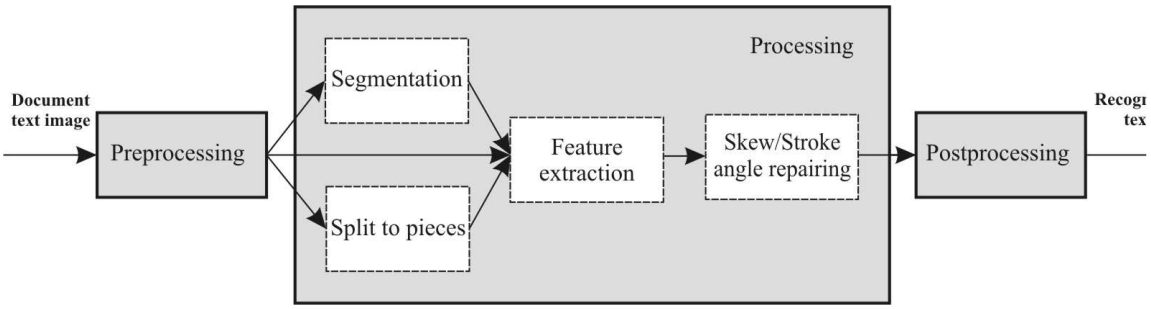

Fig. 1. Document text image identification procedure

2.1. Gaussian window algorithm

Algorithm using Gaussian window (GW) expands black pixel area by scattering every black pixel in its neighbourhood. Around every black pixel new pixels are dispersed. Those pixels have lower intensity of black i.e. level of greyscale. Its intensity depends on their position or distance from original black pixel. Our document image matrix is again greyscale. Hence, intensity pertains in level region [0-255]. Our black pixel of interest has coordinate $X_{i,j}$ and intensity of 255, while neighbour pixels have around coordinates and intensity smaller than 255 i.e. greyscale level.

After applying Gaussian window, equal to $2*K+1$, on document image, text is scattered forming enlarged area around it.

Converting all non black pixels in the same area, as well as inverting image, forms the black pixel expanded areas. Expanded areas example is given in Figure 2.

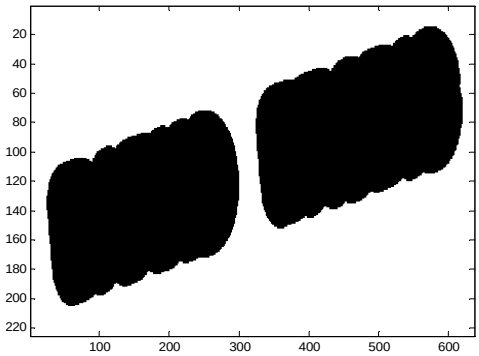

Fig. 2. Expanded text areas

2.2. Water flow algorithm

Original water flow algorithm (WF) assumes hypothetical water flows under only few specified angles of the document image frame from left to right and vice versa [1]. Previously, the definition of pixel type is needed (See Figure 3).

	i	i+1
j-1	o	o
j	x	o
j+1	x	o

a)

	i	i+1
j-1	x	o
j	x	o
j+1	o	o

b)

	i	i+1
j-1	o	o
j	x	o
j+1	o	o

c)

Fig. 3. a) Upper boundary pixel, b) Lower boundary pixel, c) Boundary pixel for additional investigation

Proposed algorithm verifies boundary pixel type in document text image. After verification, it makes unwetted areas around the words. Due to upper or lower pixel type, area slope is α or $-\alpha$. Specifically, additional verification is made on pixel for additional investigation. It can be lower, upper or no boundary pixel due to its neighbour area. Apart from [9] and [10] enlarged window R x S pixels is defined as a basis. For analysis, it is proposed $R = 5$ and $S = 7$ [6]. Position of window is backwards from pixel candidate for additional investigation. After additional investigation pixel type is designated [6]. Simplifying, unwetted areas algorithm draws area under specified angles. As a result words are bounded by unwetted dark stripes. These regions are pointed out by lines defined as:

$$y_\alpha = k * x , \qquad (4)$$

where slope $k = \tan(\alpha)$. Lines defined by slope make connection in specific pixel creating unwetted area defined as grey region in Figure 4.

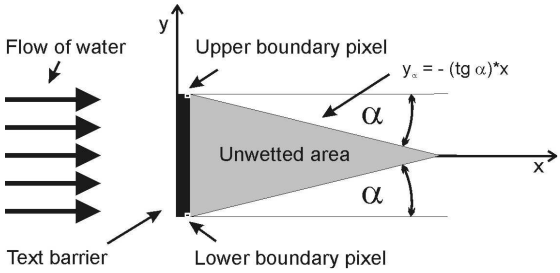

Fig. 4. Expanded text areas

Basic water flow algorithm is proposed with fixed water flow angles of: 14º, 18.4º, 26.6º and 45º applying distinct masks on original document image [1]. In [6] water flow algorithm is extended in its formulation. Still, making straight lines from boundary pixel type and connecting each others in specified point makes unweted region as well. But, water flow algorithm is free to choose different α from 0 º to 90º. Unfortunately, whole range of α can't employ due to words limitation to form connected text line regions.

2.3. Reference text line calculation

The reference line and skew angle identification is based on information obtained from black pixel expanded areas after applying algorithm. Created areas are corner stone of reference text line calculation, estimation and extraction. Defining reference text line means calculating specific average position

of only black pixels in every column of document image. Calculating reference text line is given by:

$$X_i = \frac{\sum_{j=1}^{L} Y_j}{L} \qquad i = 1, \ldots, K \ , \tag{5}$$

where X_i is point position of calculated reference text line, i is number of column position of calculated reference text, Y_j is position of black pixel in column j and L is sum of black pixel in specified column j of an image [1, 9, 10].

After calculation, image matrix with only one black pixel per column is obtained. Black pixel per column defines calculated reference text line and text line skewness. "Calculated" reference text line forms continuous or discontinuous line partly or completely "representing" reference text line. To form continuous reference text line from point's collection some numerical method could be used.

3. TEXT EXPERIMENTS

3.1. Printed text experiment

For the first experiment, sample printed text no.1 is rotated up to 45° by step of 5° around x-axis. Sample text no.1 is given in Figure 5. This sample text reference line is represented by:

$$y = a * x + b \ , \tag{6}$$

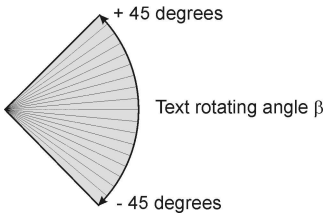

Fig. 5. Sample text no.1

After applying algorithm to sample text, reference text line is calculated by (5). To achieve continuous linear reference text line from point's collection, least square method is used. First degree polynomial function approximation is given by:

$$y = a' * x + b' , \tag{7}$$

ndp (number of data points) is used and the slope a', and the y-intercept b' are calculated as [10]:

$$a' = \frac{(\sum y)*(\sum xy) - ndp*(\sum xy)}{(\sum x)^2 - ndp*(\sum x^2)} \ , \tag{8}$$

$$b' = \frac{(\sum x)*(\sum xy) - (\sum y)*(\sum x^2)}{(\sum x)^2 - ndp*(\sum x^2)} \ . \tag{9}$$

Further, referent line hit rate (*RLHR*) is defined by:

$$RLHR = 1 - \frac{\beta_{ref} - \beta_{est}}{\beta_{ref}} \ , \tag{10}$$

where β_{ref} is arctangent of a (origin) from (6) as well as β_{est} is arctangent of a' (calculated i.e. estimated) from (7). *RMS* values are calculated by [11]:

$$RMS = \sqrt{\frac{1}{R}\sum_{i=1}^{R}(X_{ref} - X_{est})^2} \ , \tag{11}$$

where R is number of examined text rotating angles up to 45°, X_{ref} is *RLHR* for β_{est} equal to β_{ref}, due to normalization equal to 1, and X_{est} is *RLHR*.

3.2. "Handwritten" text experiment

For further experiment, "handwritten" sample text no.2 is used. Example of "handwritten" text is given in Figure 6.

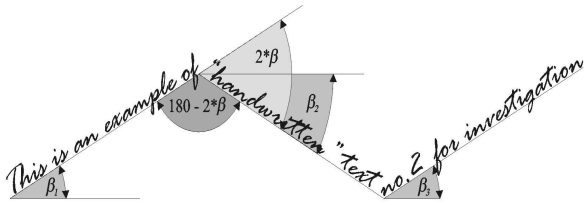

Fig. 6. Sample text no.2

$\beta_1 = \beta$, $\beta_2 = -\beta$ and $\beta_3 = \beta$ are angles of the first, second and third text line, respectively. It could be noticed, second and third text lines are rotated by $2*\beta$ from "previous" reference text line at once. Hence, this example is rather extreme one. Sample text no.2 with β from +5° to +25° by step 5° and the water flow angle α from 10° to 30° by step 5° are examined.

4. RESULTS AND DISCUSSION

GW and WF algorithms *RLHR* for sample text no.1 is given in Figure 7 and 8.

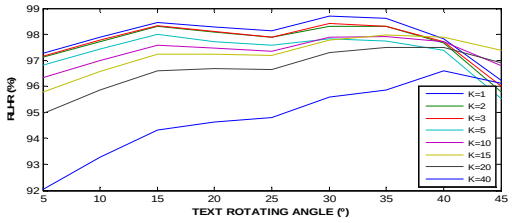

Fig. 7. Gaussian window (GW) *RLHR*

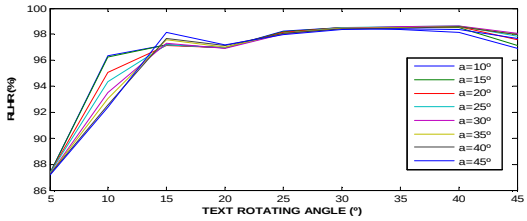

Fig. 8. Water flow (WF) *RLHR*

GW *RLHR* is in region 92%-98.5%. Using GW parameter *K* between 10 and 20 is smart enough due to better segmentation characteristics. This led to *RLHR* of 95%-97%. WF *RLHR* is in region 88%-98.5%. Using water flow angles bigger then 15° led to *RLHR* between 96%-98%. WF vs. GW algorithm *RMS* comparison is given in Figure 9.

Using bigger *K* in GW or bigger α in WF tend to wider deviation of results i.e. greater *RMS*. WF is superior in text rotation angles sub region 5°-25°, but GW is better in whole region.

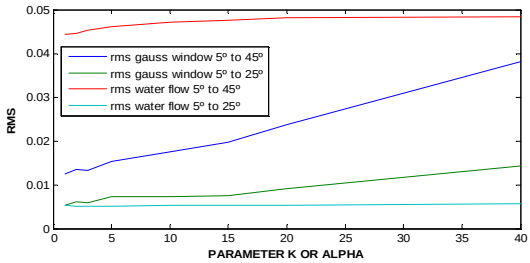

Fig. 9. *RMS* algorithm comparison

GW and WF *RMS* for fractured sample text no.2 is given in Figure 10 and 11.

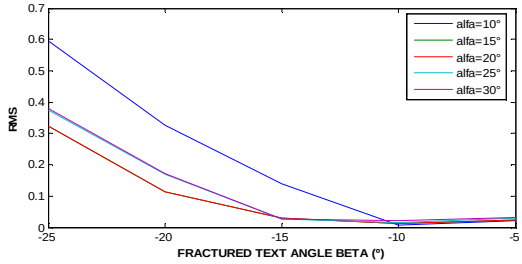

Fig. 10. Fractured text WF *RMS*

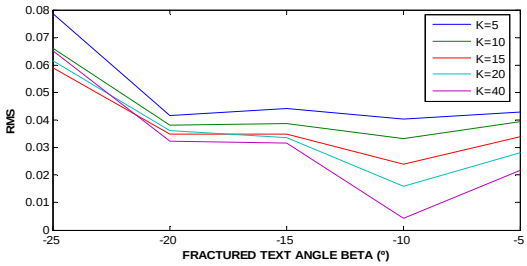

Fig. 11. Fractured text GW *RMS*

For β_i, $(i = 1, 2, 3)$ up to 15° WF is still usable as such as GW. Insensibility to errors is strength of GW evident for bigger β_i. GW and WF fractured text mean *RLHR* of β_i is given in Figure 12 and Figure 13.

Still, previous noting is similar i.e. WF is better for smaller text rotation angle and GW is better for wider text rotation angle variation.

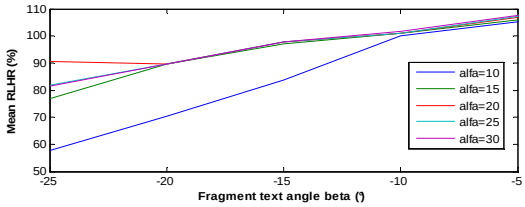

Fig. 12. Fractured text WF mean *RLHR*

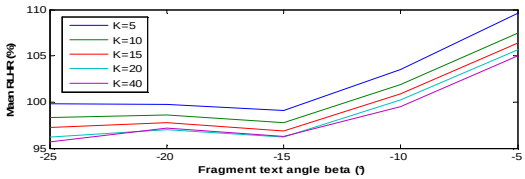

Fig. 13. Fractured text GW mean *RLHR*

5. CONCLUSION

In this paper, algorithms for reference text line and skew rate identification of printed and handwritten text is presented. It assumes creation of expanded text area based on water flow or Gaussian window algorithm. Both algorithms are analyzed and examined under printed and "handwritten" text samples. Water flow algorithm is well behaved for "smaller" text rotation angle up to 30°, while Gaussian window is more robust in wider rotation angle region. Further improvement of water flow algorithm should be made in creating "water" to follow text rotation angle.

References

[1] S. Basu, C. Chaudhuri, M. Kundu, M. Nasipuri, D. K. Basu, "Text Line Extraction from Multi-Skewed Handwritten Documents", *Pattern Recognition*, Vol.40, pp. 1825-1839, 2006

[2] A. Amin, Sue Wu, "Robust Skew Detection in mixed Text/Graphics Documents", *International Conference on Document Analysis and Recognition (ICDAR'05)*, 2005

[3] G. Louloudis, B. Gatos, I. Pratikakis, C. Halatsis, "Text Line Detection in Handwritten Documents", *Pattern Recognition*, Vol.41, pp. 3758-3772, 2008

[4] Yi Li, Yefeng Zheng, D. Doermann, S. Jaeger, "A New Algorithm for Detecting Text Line in Handwritten Documents", *18th International Conference on Pattern Recognition*, Vol.2, pp. 1030-1033, Hong Kong, 2006

[5] Jiren Wang, Mazlor K.H. Leung, Siu Cheung Hui, "Cursive Word Reference Line Detection", *Pattern Recognition*, Vol.30, No.3, pp. 503-511, 1997

[6] D. Brodić, Z. Milivojević, "Reference Text Line Identification Based on Water Flow Algorithm", *ICEST '2009*, SP-2 Sect., Veliko Tarnovo, Bulgaria, 2009

[7] D. Brodić, Z. Milivojević, "Using Gaussian Window for Printed and Handwritten Text Parameters Extraction", *BALCOR '2009*, Constanca, Romania, 2009

[8] I. V. Draganov, A. A. Popova, "Rotation Angle Estimation of Scanned Handwritten Cursive Text Documents", *ICEST '2006*, Sofia, Bulgaria, 2006

[9] R. C. Gonzalez, R. E. Woods, *Digital Image Procesing*, 2nd ed., New Jersey: Prentice-Hall, 2002, pp. 67-70

[10] M. Sonka, V. Hlavac, R. Boyle, *Image Processing, Analysis and Machine Vision*, Toronto: Thomson, 2008, pp. 174-177

[11] W. M. Bolstad, Introduction to Bayesian Statistics, New Jersey: John Wiley & Sons, 2004, pp. 40-44, 235-240