Text Regions Segmentation in Image Printed Documents

Antoaneta A. Popova[1], Milen A. Dimitrov[2], Vasil G. Grancharov[3]

Abstract – **The paper deals with a suitably developed algorithm and designed program system that is being used to separate text regions from image and graphics regions. A combination is used of an edge detection, a dilatation, a recognition text blocks by features as a periodicity of 90^0, an uniformly/ regularity of 0^0 and a similarity of 45^0 and 135^0 projections.**

Keywords – **Page segmentation, Text extraction, Edge detection, Document image understanding.**

I. INTRODUCTION

The image documents analysis is part of the conversion task from a hard copy to a soft copy with text recognition in the office automation. The goal of a document image understanding system is to convert a raster image scanned by a document scanner, into an appropriate symbolic form [1]. One of the applied approaches is a morphological closing or only erosion on the different directions (60°, 120°), using a structural element of 5x5 in order to eliminate non-text objects [2]. The X-Y tree method (pyramid) assumes that a document can be represented in the form of nested rectangular blocks. A "local" peak detector is applied to the horizontal and vertical "projection profiles" to detect local peaks [3]. In the above and most of the previous works [4,5,6] the features for block classifications are extracted from the binarized document images. This has some difficulties such as the determination of the threshold value for the binarization and the lack of enough information for detailed block classification. So, for an efficient block classification, the feature extraction from gray images is applied in this work. In Section 2, is described in detail the approach for automatic text segmentation algorithm from gray-scaled images. Experimental test results are presented in Section 3. Finally, the conclusions are given in Section 4.

[1]Antoaneta A. Popova is with the Faculty of Telecommunication, Technical University- Sofia
Bulgaria, E-mail: Antoaneta.P@komero.net
[2]Milen A. Dimitrov is with Komero Technolodgies Int. Ltd., Sofia, Bulgaria, E-mail: Milen.D@komero.net
[3]Vasil G. Grancharov is with Orbitel Ltd., Sofia, Bulgaria

II. TEXT SEGMENTATION ALGORITHM

The function of the segmentation algorithm is to locate the information blocks in the document image. Two approaches, *boundary-based* (edge information transition from background to the information object block) and *region-based* (fill-up and segment these blocks).

Fig. **1**. illustrates the flow diagram for the segmentation algorithm.

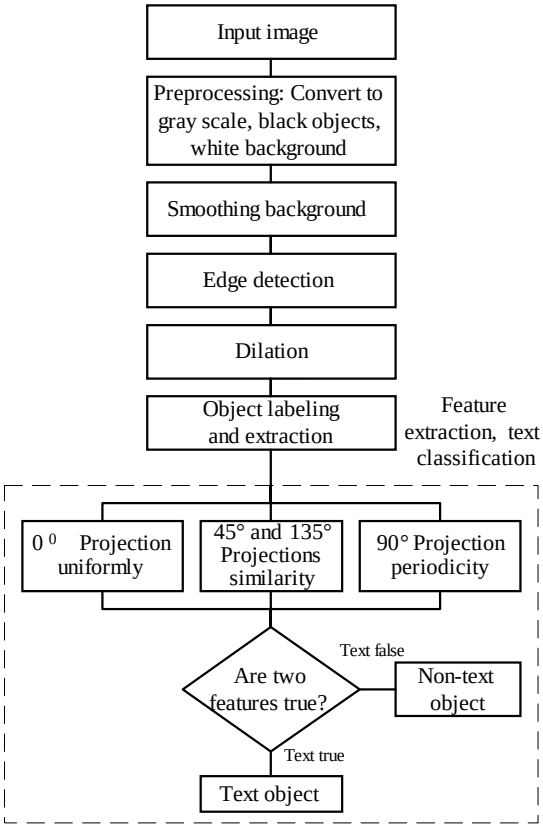


Fig. 1. Flow diagram for segmentation algorithm

Smoothing Background: This operation removes slight variation in the background by the histogram *h(i)* analysis. The threshold is set to retain 18% of pixels with the lowest intensities. Pixels *Z(i)* with intensities *i* above the threshold are set to 255 (background intensity level). If in the intensity interval 254-127 there isn't 18 % pixels all pixels in this interval are set to 255.

For k=0 to 127 $If \sum_{i=254}^{254-k} h(i) = (18 * \sum_{i=0}^{255} h(i))/100$ (1)

$Z(i)_{254-k}^{254} = 255$

$If \sum_{i=254}^{127} h(i) < (18 * \sum_{i=0}^{255} h(i))/100$ (2)

$Z(i)_{254}^{127} = 255$

Edge-detection: The edge detector seeks for gradient transform of the gray scale image *A*, using the Prewitt method. It realizes a convolution of two filter masks horizontal *Px* and vertical *Py* (Fig. 2.). The current pixel *Z8* is in the center of the window operator *A1-A9*.

-1	0	1
-1	0	1
-1	0	1

-1	-1	-1
0	0	0
1	1	1

A1	A2	A3
A4	A5	A6
A7	A8	A9

Fig. 2. Prewitt filter masks *Px, Py* and image area *A*

As the convolution result of the gray scale in the current pixel and his neighbors with masks *Px*, *Py* are obtained two scalar components *Hx*, *Hy*. The image area *A* is considered, when *A1..A9* are gray scale values. *Hx* and *Hy* are calculated by multiply the image area with *Px* и *Py*.

$$Hx = (A3 + A6 + A9) - (A1 + A4 + A7) \quad (3)$$

$$Hy = (A7 + A8 + A9) - (A1 + A2 + A3) \quad (4)$$

The magnitude *H(x,y)* and orientation *Q(x,y)* of the gradient can be computed from the standard formulas for rectangular-to-polar conversion:

$$H(x,y) = \sqrt{H_x^2(x,y) + H_y^2(x,y)} \quad (5)$$

$$Q(x,y) = \mathrm{Arctan}(H_y / H_x)$$

Then is calculated *H(x,y)*, and compared with a threshold 100. If *H(x,y)* value is less or equal to threshold the current pixel is set 0 (black) or in the other case is set 255 (white). In this way is done edge detection and binarization.

Dilation: A square structuring operator 3x3 is used. The dilation/ closing operation aims to fill-up the region and increase the connectivity within pixels so that labeling can be done more accurately. A square-structuring element is therefore a suitable candidate. The dilation is done repeatedly until there is no change in the output of this operation. If the current pixel is 0 and there is at list one neighbor with intensity *i=255* in 3x3 matrix the current pixel is set to 255.

In this method is applied the morphological nonlinear operation dilation on the binary page document image. The dilation operation $\otimes$ combines two sets *A* and *B*, using a vector sum. Dilation $A \otimes B$ is pixels set of all possible vector sums of couple pixels, by one from each pixel sets *A* and *B*.

$$A \oplus B = \{p \in \varepsilon^2 : p = a + b, a \in A, b \in B\} \quad (6)$$

Using structural operator *B* 1x2 is obtained the original object *A* closing on horizontal direction. The dilation example is given on (
Fig. **3**.), and with *X* is marked the coordinate system origin.

$$A = \{(1,0),(1,1),(2,1),(1,2),(3,2),(1,3),(4,0)\} \quad (7)$$

$$B = \{(0,0),(1,0)\} \quad (8)$$

$$A \otimes B = \{(1,0),(1,1),(2,1),(1,2),(3,2),(1,3),(0,4),(1,4),(2,3),(2,2),(4,2),(3,1),(2,0)\} \quad (9)$$

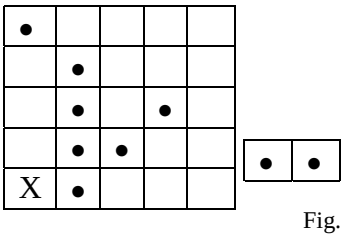
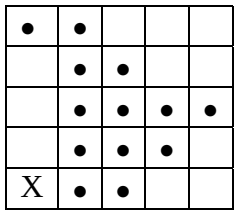


Fig. 3. Dilation: original object, operator, closing object

Objects labeling and extraction: A connected component is a set of connected pixels that share a specific property *V* (same color, intensity). Two pixels *p* and *q* are connected if there is a path from *p* to *q* of pixels with property *V*. The image will be represented by an array *A* that has *N* columns and *M* rows.



A[*x,y*] refers to the element in column *x* and row *y*, with $x \in \{0, N-1\}, y \in \{0, M-1\}$. Let *Q* be an array that is the same size as *A* and will hold the connected component labels *L*. For the beginning *L*=0 and is incremented for a new connected component. The goal is to end up with all of the pixels in each connected component having the same label *L* and all of the distinct connected components having different labels *L*. Some of the pixels in the same connected component will end up with different labels and it will be resolved at the end of the labeling process. A vector EQ will hold the equivalence class relations that are discovered as the algorithm is running.

The pixels with value zero are image background. The following algorithm will be described for 4-connected neighborhoods, applied on the binary image.

Step 1: Label pixel A[0,0]. If *A*[0,0]>0 then increment *L* and set *Q*[0,0]=*L*. This takes care of the first pixel in the image.

Step 2: Label the pixels in row y=0. For *x*=1 to *N*-1, check the value of *A*[x,0]. If *A*[x,0]>0 and *A*[x,0]=*A*[x-1,0] then set *Q*[x,0]=*Q*[x-1,0]. If *A*[x,0]>0 and $A[x,0] \neq A[x-1,0]$ then increment *L* and set *Q*[x,0]=L. This will cause

neighboring pixels in the first row that have the same value to have the same label (Fig. 4.).

0	1	1	1	1	0	0	2	2	2	2	0	0	0	3	3	0	0	4	4

Fig. 4. Sample of the first row pixels labeling

Step 3: Label the rest of the rows. The first element of each row is handled a little differently than the rest of the elements, since it has no left neighbor. For y=1 to M-1 do the following. If $A[0,y]>0$ and $A[0,y]=A[0,y\text{-}1]$ then set $Q[0,y]=Q[0,y\text{-}1]$. If $A[0,y]>0$ and $A[0,y] \neq A[0,y\text{-}1]$ then increment L and set $Q[0,y]=L$. This takes care of labeling the first element in the row.

Labeling the rest of the elements in a row requires that we look at both the neighbor above and the neighbor to the left. For simplicity, let us refer to the current pixel as p, the neighbor to the left as s and the one above as t. If $A[p]=A[s]$, but $A[p] \neq A[t]$ then set $Q[p]=Q[s]$. If $A[p] \neq A[s]$, but $A[p]=A[t]$ then set $Q[p]=Q[t]$. If $A[p] \neq A[s]$ and $A[p] \neq A[t]$, then increment L and set $Q[p]=L$. If $A[p]=A[t]$ and $A[p]=A[s]$ and $Q[s]=Q[t]$, then set $Q[p]=Q[t]$. This takes care of all the cases except if $A[p]=A[t]$, $A[p]=A[s]$ and $Q[s] \neq Q[t]$. This means that pixels s and t have the same values but different labels, and they are in the same component. We therefore label pixel p with the smaller of the two labels and record the fact that the larger label value is equivalent to the smaller one. Let $L1$ be the smaller value and $L2$ be the larger value, then set $Q[p]=L1$ mark $EQ[L2]=L1$. The first three rows are shown in Fig. 5.

0	1	1	0	0	0	2	0	0	3
0	0	1	0	4	4	2	0	5	3
0	0	1	0	0	4	0	0	0	0

Fig. 5. The first three image rows labeling

Features extraction and text classification: The purpose is to segment the image into text and non-text regions as best as possible, and then let the OCR system do make symbol recognition. The used features are the following:

0^0 Projection uniformly/ regularity feature: The most intuitive characteristics of text are its regularity. If the average of each three neighbors projection values are approximately equal (small dispersion) the regularity/ uniform feature test is true/ positive.

45^0 and 135^0 Projections similarity feature: The difference between the autocorrelation of 45^o projection and the cross-correlation between 45^o and 135^o projections is first computed.

The used equation for the cross-correlation normalized function is:

$$r = \frac{\sum_m [(x(m) - x_{aver}) * (y(i-d) - y_{aver})]}{\sqrt{\sum_m (x(m) - x_{aver})^2} \sqrt{\sum_m (y(m-d) - y_{aver})^2}} \quad (10)$$

where x,y are the projection pixel coordinates and i, m are projection elements numbers. If $r>=0,90$ can be accepted that these projections similarity is a positive feature.

90^0 Projection periodicity feature: The peak detection is carried out. Next the intervals between the peaks are extracted and the mean value of the interval calculated. If there are more than 50% of the intervals with regular spacing, the test is positive.

Our system takes advantage of the distinctive above characteristics of text that make it stand out from other image material. The text classification in two classes text and non-text is done if two of the above extracted features are true.

III. Experimental Test Results

The segmentation algorithm was tested on digitized paper documents. The proposed method was tested on 8 newspaper images with 23 separated text, images and graphics objects. Image size varied to 1024-768 pixels and 256 grey scale levels. Most of the text blocks were successfully separated from non-text regions. The algorithm for locating text is relatively fast.

Details of the implementation: The results obtained for this segmentation algorithm are shown in the below figures.

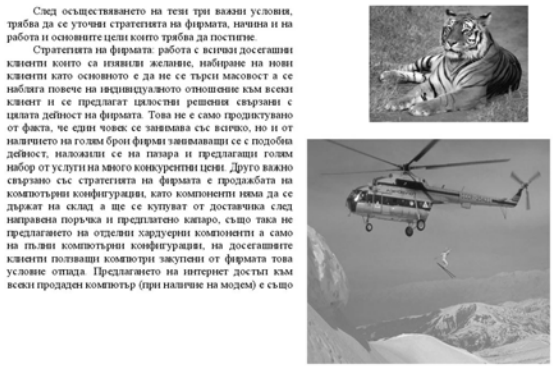


Fig. 6. Input original document image with text and plane

Fig. 7. After smoothing background and edge detection

Fig. 8. After dilation for connected component analysis

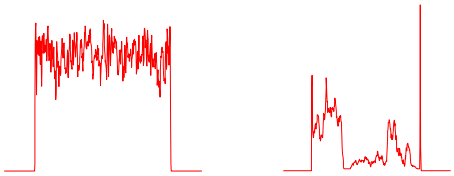


Fig. 9. 0^0 Projection uniformly feature (text, plane)

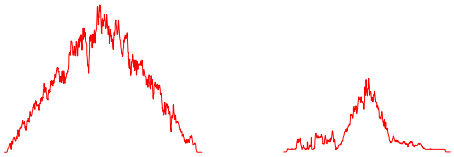


Fig. 10. 45^0 Projections for similarity feature (text, plane)

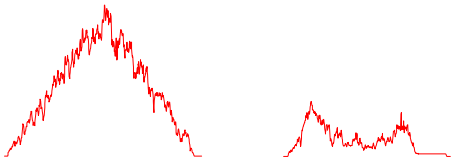


Fig. 11. 135^0 Projection for similarity feature (text, plane)

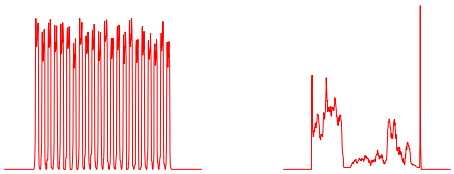


Fig. 12. 90^0 Projection periodicity feature (text, plane)

The developed algorithm was implemented as a Delphi program and tested in a Laboratory Television, Faculty of Communication and Technologies, Technical University-Sofia. Two computers' configurations were used under MS Windows 2000: Celeron 1700/ 500 MHz, RAM 512/ 128 MB. The final performance result for both configurations was less than one second. This good performance due to using for image manipulations "memory bitmap"and "scanline".

IV. Conclusions

In conclusion, a text segmentation algorithm is developed in two steps, segmentation and text classification. The two steps are implemented and tested. The programming realized algorithm is independent of the size and type of the characters and the position of the text in the document. This paper proposes a segmentation method that clusters successfully the regions of a mixed-type document image into text for OCR input and non-text areas.

There are some difficulties if the text object has only one row and the feature *Periodicity* 90^0 will be negative. This text region can be mixed-up with some texture region. In the future this research will be extended with exploring of the 0^0 projection for periodicity corresponding to the symbols' columns and space between them, using on word identification to make the algorithm more robust and suitable to different cases. A weighting function can be implemented instead of the hard AND operation in the final output of classification.

Reference

[1] S. Srihari, S. Lam, V. Grovindaraju, R. Srihari "Document image understanding", CEDAR, State University of New York at Buffalo, 1999.

[2] C. Ngin, "Text segmentation in printed documents", Computer Society Press, EE368A Final Project Report, 2001.

[3] "Project DEBORA", Instituto Superior Técnico (PT), 1999.

[4] Q. Yuan, C. Tan, "Page segmentation and text extraction from gray scale image in microfilm format", National University of Singapore, 2002.

[5] V. Wu, "Finding text in images", Computer Science Department, University of Massachusetts, 2001.

[6] R. Gonzalez, R. Woods, "Digital Image Processing", A-W Publishing, 1993.