ISSN:2231-0711
www.ijcset.net

Word Segmentation for Document Images by Successively Merging Adjacent Character Bounding Boxes by Iterative Dilation

Shashidhara B and Bhaskara Rao. N.

Dayananda Sagar College of Engineering,Dept. CSE, Bangalore. India.
bolasasi@gmail.com
bhaskararao.nadahalli@gmail.com

Abstract— **A new method of word segmentation for document images is presented. The method uses the bounding box regions to enclose the letters (characters) of the words and then the resulting letter spaces are progressively filled to merge the character bounding boxes to get the word bounding boxes. The method holds good for inclined and irregularly distributed words. The proposed method completely avoids the line segmentation process which normally precedes word segmentation in traditional methods.**

Keywords— **Bounding boxes, Connected components, Horizontal Dilation, Character spacing, Word bounding boxes, Word segmentation, Word spacing.**

1. INTRODUCTION

Word segmentation, that is, the separation of individual words of a text in a given document image, is an important task in document analysis and OCR system [1]. There are several existing methods for word segmentation. They follow either the top-down approach or the bottom-up approach [2]. In our method, we use the bottom-up approach to segment the words starting from individual characters (letters) of the text.

II. PROBLEM STATEMENT

In those situations, where line segmentation is not essential and word segmentation is sufficient, we can go only for word segmentation. Our problem is to get the word segmentation of a given text document image, without involving or invoking the line segmentation.

III. PRE-PROCESSING

The information content of a text document image can be easily represented in its black-and-white format. Therefore, if the given document image is in RGB or gray scale format, it is converted to its black-and-white format for further processing. Also, we assume that the document image is noise free, skew free and well formed and there are no broken and degraded characters. If so, the image is pre-processed to remove the noise, line skew and to correct the broken and degraded characters. In our method, we apply morphological transformations on to the target image. Therefore we need text characters as white pixels objects with black background. Hence, if the black-and-white document image has black characters on white background, as generally found, it is inverted to get white characters on black background.

IV. PROPOSED METHOD

Here we follow the bottom-up approach for word segmentation starting from characters. The first step is to detect and mark the individual characters.

A. Connected Component Labeling

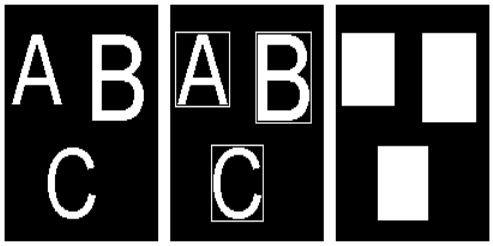
English and other European languages use the roman script. Here, except for a few special cases, an individual letter (character) is a single connected object. That is, the pixels forming a single character are all connected. Exceptional cases are, letter **i** which has two components and special characters like colon, semicolon, double quote, question and exclamatory marks which also have two components. The % symbol has 3 components. However, in these special cases the sub components of the character are very close to each other and can be easily merged into a single component. Similar argument holds good for most of the other languages. Thus, to access individual characters, we use 8-connected component labeling. Matlab Image Processing Toolbox (IPT) function **bwlabel** [3] is used to generate all the connected components of the binary image. Then the connected components are marked by the respective rectangular bounding boxes which just enclose them completely.

B. Character Bounding Box format

Bounding Boxes [4] for connected components are the properties of the labeled connected component regions. A bounding box of a labeled region is a rectangle that just encloses the region completely. When a specific bounding box is determined for a connected region, the co-ordinates of the corners of the bounding box and its width and height are available. A bounding box completely specifies the boundaries of the corresponding connected component. In our method, we use filled bounding boxes which completely cover the corresponding connected components. Fig.1 shows the bounding boxes which enclose their connected components. Fig.1(a) shows the document image. Fig.1(b) shows the unfilled bounding rectangles (boxes) while Fig.1(c) shows the same rectangles (boxes) filled with white pixels.

The Bounding Box for a connected component is determined using the Matlab IPT function *regionprops* [4]. Functions *bwlabel* and *regionprops* work correctly when the foreground objects of the binary image are

white and the background black. In Fig.1, the text object is taken white (pixel value =1) while the background is black (pixel value = 0).



(a) (b) (c)

Fig.1. A document image and its Bounding Boxes
(a) Document Image
(b) Unfilled Bounding Boxes
(c) Filled Bounding Boxes

C. x gaps, character gaps and word gaps

The horizontal spatial separations between adjacent bounding boxes are measured parallel to the x-axis and hence called x gaps. Some of these x gaps are the horizontal separating gaps between the corresponding adjacent characters (letters) and the remaining x gaps are the adjacent word gaps of the document image. Word gaps are the separation between the adjacent words. The word gap between two adjacent words is the gap between the last character of the first word and the first character of the second word. An example of a 3 word document image is shown in Fig.2. In Fig. 2(c), the filled bounding boxes show that it has 9 x gaps, 2 word gaps and 7 character gaps. The first word gap occurs between characters 4 and 5 and the second gap between 7 and 8. All other gaps are character gaps. In general, word gaps are much wider than character gaps. Our objective is to distinguish between the character gaps and word gaps and to fill the character gaps without filling the word gaps.

D. Separation of word gaps from character gaps

In a document image, the x gaps can have varying widths. These x gaps, represented by their widths form a set. We call this, the x gap set. The x gap set is a collection of all the distinct x gap widths. Character gaps and word gaps are mutually exclusive and exhaustive sub sets of the x gap set. The character gaps have smaller widths than the word gaps. The average width of word gaps is much larger than that of the character gaps. The distribution of x gap widths is bimodal [5]. Hence from the x gap set, the character gap set (set of all letter gaps) and the word gap set (set of all word gaps) can be separated using a suitable threshold for separation.

For example, in the single line resized document image of Fig.2, the x gaps in pixels are as shown below.
All x gaps from left to right = [12 9 7 33 9 8 31 8 8]
After deleting the duplicate entries and arranging the elements in the increasing order, the x gap set is,
x gap set = [7 8 9 12 31 33].
After separation, the two distinct set are,
Character gap set = [7 8 9 12].
Word gap set = [31 33].

(a) Document Image

(b) Unfilled Letter Bounding Boxes

(c) Filled Letter Bounding Boxes

Fig.2. Letter gaps and word gaps for a 3 word text document image

III. DETERMINATION OF X GAPS

Determination of x gaps is the most important and innovative stage in the proposed method. The x gaps depend on the nearest horizontal adjacency and are difficult to determine without the accompanying preceding line segmentation information. In our method, the x gaps are determined directly, skipping the line segmentation process.

A. Morphological Dilate operation

This is cleverly done using the Morphological Dilation [6] operation. The morphological dilation operation expands the labeled components, based on the size and shape of the specified structuring element. When the labeled components (letter bounding boxes) expand, the intervening gaps get reduced. In the proposed method, the expansion is restricted along the horizontal direction only, so that it affects the x gaps but not the vertical separations. Also, the horizontal dilation is made unidirectional (say from left to right) so that the precise width of dilation can be controlled. This is achieved by using a special single row structuring element as described below.

B. Special Structuring element for Unidirectional Dilation

Here, the structuring element is chosen as a single row matrix of n zeros followed by (n+1) ones for n=1,2,3…etc.
Thus,
for n=1, SE_1 = [0 1 1],

for n=2, SE_2 = [0 0 1 1 1],
for n=3, SE_3 = [0 0 0 1 1 1 1].
…….
In general for a specified n,
SE_n = [zeros(1,n) ones(1,n+1)] (1)
Here, zeros(1,n) is a 1xn matrix of 0's
and ones(1,n+1) is a 1x(n+1) matrix of 1's.
The size of SE_n itself is 1x(2n+1).
The special properties of SE_n are,

1. It has only one row so that the dilation occurs only along the horizontal direction.
2. Total number of elements in it is odd and its center element is 1 with n 0's on its left and n 1's on its right so that the direction of dilation is from left to right and the width of dilation is exactly n pixels.

C. Determination of x gaps using Unidirectional Dilation

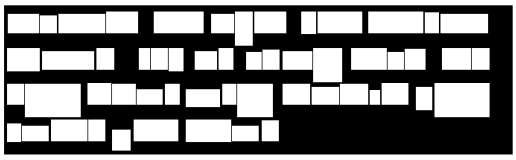
In a given bounding box document image, a unidirectional dilation operation with SE_n extends the right edges of the bounding boxes by n pixels to right. During dilation, if the right edge of a bounding box touches or crosses the left edge of its right neighbor then that x gap is filled up. A merger occurs. Let m be the width of an x gap before dilation. Let n be the width of dilation. Then,
if n<m, there is no merger after dilation. The x gap is not filled up. x gap width is reduced from m to (m-n).
If n = =m, it results in a just merger. The x gap is filled up.
If n>m, it results in a merger with overlap. The x gap is filled up.

In the bounding box image of a document, a merger during dilation can be detected by counting number of connected components before and after dilation. When an x gap is filled, a merger occurs and the number of distinct components gets reduced by 1.

Let N1 and N2 be the number of components before and after the dilation operation respectively. If N2 is equal to N1, no gaps are filled up. When N2 is less than N1, The number of connected components is reduced by N1-N2 and that many gaps are completely filled up during that dilation process. When the dilation width n (in pixels) is less than the smallest of the x gap widths, say w1, no merger takes place. Then n is progressively increased for the successive dilations. When n just reaches w1, merging occurs and the value of n determines w1. There may be more than one x gap having this w1 value. Let the number of x gaps merged when the width is w1 be denoted by nw1. This value nw1 is given by (N1-N2) where N1 and N2 are the number of components before and after the merger. The next x gap width w2 higher than w1 and the corresponding nw2 are also obtained similarly. In this way, all the remaining x gaps are obtained by the dilate operation which is successively repeated with progressively increasing dilation width. Thus the x gap-width list which contains { w1, w2, w3,…} and the corresponding number_of_x_gaps list containing {nw1, nw2, nw3, …} are determined

C. Stopping criterion

Let nmax be the maximum x gap width of a given document image. Once the dilation width n reaches nmax, all the x gaps of the bounding box image are merged. The resulting image is a series of horizontal bars with no x gaps at all. An example where all x gaps are merged is shown in Fig.3. Further dilation with higher n has no effect. Therefore the repeat iteration is stopped once n reaches nmax. An indirect way of determining nmax is to dilate the target image with a dilation width equal to that of the image itself. Then, maximum dilation occurs and all the x gaps are merged and only the horizontal bars are left out, the number of horizontal bars being the number of text lines. Let this number be NL (Number of Lines). This situation effectively is same as dilation using n = mmax. Thus the dilation iteration count is increased uniformly until the number of components remaining after the mergers is equal to NL.



(a) Bounding box image before dilation

(b) Bounding box image after full x-dilation

Fig 3. Bounding box image before and after, for a 4 line document

The get x gap algorithm with the above stopping criterion is given below. It is assumed that the filled bounding box image for a given document image is available after due pre-processing.

D. Get x gap Algorithm

Let B be the binary filled bounding box image. Let the width of the image be W. Let NL be the number of text lines.

1. With W as the width of dilation, get the full x dilation of B to get C. See Fig. 3(b).
   From C, find the number of connected components to get NL.
2. Get N1, the initial number of connected components by applying bwlabel function to B. That is obtain N1 as,

[L N1]=bwlabel(B). (Here L is not used.)

3. Initialize x_gap_width and number_of_x_gaps lists to null as,
   X_gap_width=[],
   number_of_x_ gaps=[].
4. Start with n =1 to set the dilation width to 1.
5. While N1>NL
6. Construct SE_n according to Eq.(1).
7. Do morphological dilation as,
   C = imdilate(B,SE_n).
8. Get N2, the number of connected components after dilation as,
   [L N2]=bwlabel(C).
9. Compare N2 with N1.
   If (N1-N2) ==0, (there is no merger.)
   Increment n by 1 as, n=n+1 and go to step 5.
   else,
   A merger has occurred and the present dilation
   width n is one of the x gap widths.
   Append this n into the x_gap_width list as,
   x_ gap-width=[x_ gap_width n] and append the
   number_of_ x_gaps list as,
   number_of_x_gaps=[number_of_x_gaps (N1-N2)].
   Update N1 for the next iteration as, N1=N2.
   Increment n by 1 as, n=n+1 and go to step 5.
10 End of while loop.

E. Distribution of x gap widths.

The number_of_x_gaps list gives the number of x gaps whose corresponding widths are present in the x_gap_width list. x_gap_widths and the corresponding number_of_ x_gaps for a typical document image are shown in Table I.

TABLE I. x gap_widths and their Frequencies

x_gap_width	Number_of_x_gaps (Frequency)	Δ (x_gap_width) Change in gap width
1	143	1
2	23	4
6	12	1
7	15	1
8	15	1
9	7	1
10	3	1
11	8	1
12	4	1
13	3	1
14	2	

The bar graph plot of number_of_x_gaps versus the widths from the x_gap_width list will show the x gap width distribution. The corresponding histogram plot is shown in Fig.4. The bimodal nature of the distribution can be clearly seen in Fig.4. In this example, *upper letter gap width* is 2 and the *lower word gap width* is 6. The letter mode and the word mode regions can be easily distinguished because of the clear separation between the two. In this example, the separating point is at width=4 (see Fig.4).

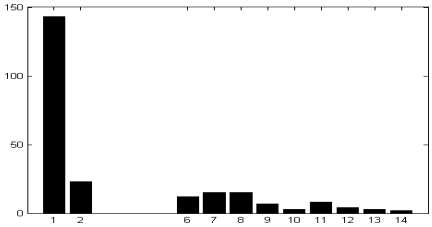
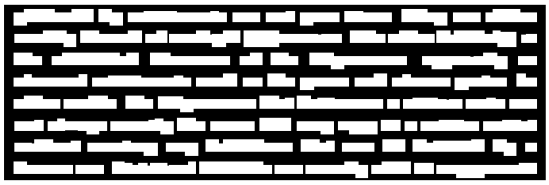
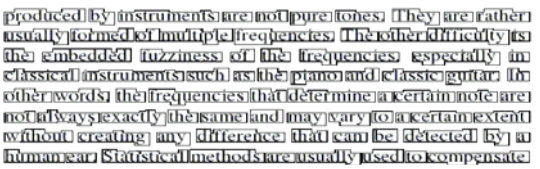


Fig.4. Bimodal distribution of x gap widths

The left to right dilation of the target image with this width will merge all the letter spaces while retaining word boundaries. The result is shown in Fig.5.



(a) Word Segmentation blocks



(b) Segmented words of the text

Fig. 5. Word Segmentation blocks and its text

In general the x gap histogram has a zero value region between the *upper letter gap width (ULG)* and the *lower word gap width (LWG).* In this region, the x gap width abruptly changes from ULG to LWG. This is determined by examining the Δ (x-gap-width) list which gives the successive changes in x gap width. The change is maximum at the transition from ULG to LWG. See column three of Table I. Smoothened histogram and other methods also can be used to determine the separating threshold [7].

IV. CONCLUSION

In this paper a new method of word segmentation of a document image has been described. Here, the line segmentation is not a pre requirement.

REFERENCES

[1] Nallapareddy Priyanka, Srikanta Pal and Ranju Mandal, " Line and Word Segmentation Approach for Printed Documents', *IJCA Special Issue on Recent Trends in Image Processing and Pattern Recognition,* RTIPPR, 2010, November, Article 4. pp 30-35.

[2] S. Chen, R. M. Haralick, and I. Phillips, "Simultaneous word segmentation from document images using recursive morphological closing transform", *Proceedings of the 3rd ICDAR*, pages 761–764, Aug. 1995.

[3] www.mathworks.com/helptoolbox/images/ref/bwlabel/html

[4] www.mathworks.com/helptoolbox/images/ref/ Regionprops.html.

[5] D. J. Ittner and H. S. Baird, "Language-free layout analysis", *Proceedings of the 2rd ICDAR*, pages 336–340, Oct. 1993.

[6] R.C. Gonzalez and R.E. Woods, Digital Image Processing, Ch.9, Pages 655-657. 3rd ed, Pearson Education. 2009.

[7] M. Sezgin and B. Sankur , "Survey over image thresholding techniques and quantitative performance evaluation', *Journal of Electronic Imaging 13(1), 146–165* (January 2004).