Research Journal of Applied Sciences, Engineering and Technology 4(24): 5438-5442, 2012
ISSN: 2040-7467

Submitted: March 18, 2012 Accepted: April 14, 2012 Published: December 15, 2012

An Efficient Character Segmentation Based on VNP Algorithm

S. Chitrakala, Srivardhini Mandipati, S. Preethi Raj and Gottumukkala Asisha
Department of Computer Science and Engineering, Easwari Engineering College, Ramapuram, Chennai 600089, India

Abstract: Character segmentation is an important preprocessing stage in image processing applications such as OCR, License Plate Recognition, electronic processing of checks in banks, form processing and, label and barcode recognition. It is essential to have an efficient character segmentation technique because it affects the performance of all the processes that follow and hence, the overall system accuracy. Vertical projection profile is the most common segmentation technique. However, the segmentation results are not always correct in cases where pixels of adjacent characters fall on the same scan line and a minimum threshold is not observed in the histogram to segment the respective adjacent characters. In this study, a character segmentation technique based on Visited Neighbor Pixel (VNP) Algorithm is proposed, which is an improvement to the vertical projection profile technique. VNP Algorithm performs segmentation based on the connectedness of the pixels on the scan line with that of the previously visited pixels. Therefore, a clear line of separation is found even when the threshold between two adjacent characters is not minimal. The segmentation results of the traditional vertical projection profile and the proposed method are compared with respect to a few selected fonts and the latter, with an average accuracy of approximately 94%, has shown encouraging results.

Keywords: Dissection based segmentation, optical character recognition, segmentation, vertical projection profile .

INTRODUCTION

Character segmentation is the decomposition of an image into its subcomponent where each subcomponent contains individual character image. It is the most critical step in many image processing applications that involve recognizing characters-like OCR and License plate recognition as it can cause high errors in the recognition stage. Many character segmentation techniques exist, but each with its own restrictions. The most commonly used character segmentation technique is vertical projection profile, which uses inter-character white space as the criteria for segmentation. However, when pixels of adjacent characters lie in the same vertical scan line, vertical projection profile treats these adjacent characters as the same character, hence segmenting the character incorrectly. In this study, Visited Neighbor Pixel (VNP) algorithm based character segmentation is proposed. The extent of each character is determined by considering the adjacency or neighborhood of two pixels, overcoming the shortcoming of the vertical projection profile technique. The connectedness of pixels is used as a factor to describe the characters. A few studies in this area are explained below.

Jaedo *et al*. (2009) proposed a robust license plate detection method which uses topological transform and perspective projection profile for character separation. The topological transform fails to find closed loops which form the areas of input containing the number plate when the intensity value is high and the edge detection does not produce accurate results. Due to this 57% of errors occur in the character detection module.

Rajiv and Amardeep (2008) proposed a method for the segmentation of handwritten text in Gurmukhi script. The segmented word is scanned vertically from left to right by ignoring the horizontal line with maximum intensity. The presence of one or more black pixels marks the beginning of the character. As the scan progresses, the location where no black pixels are encountered marks the end of the character. In this manner, the starting and ending location of each character is stored and segmented. However, in cases where the horizontal head line of a word has a break, the system fails to identify it as a single word. Problem also arises at times when the maatra of a character is not connected to it and the character segmentation module treats them as separate characters.

Min-Chul *et al*. (1999) proposed a method which uses the side profiles for the segmentation and recognition of machine printed touching characters. Seong-Whan *et al*. (1996) proposed a two-stage character segmentation method. Pre-segmentation points are first selected from the projection profile. A multi-stage graph is then

Corresponding Author: S. Chitrakala, Department of Computer Science and Engineering, Easwari Engineering College, Ramapuram, Chennai 600089, India

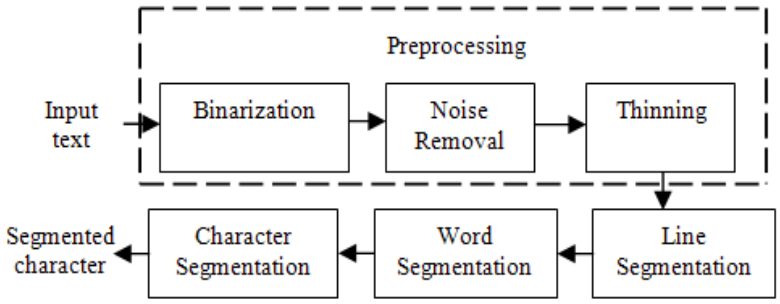

Fig. 1: Functional architecture of the system

generated to identify a more accurate character segmentation region. Sanjeev and Sudhaker (2006) also proposed a two-stage character segmentation technique, but it was tailored for printed Kannada scripts. The subscripts are first segmented from the word after which the rest of the characters are segmented using global vertical projection profile. This method has a disadvantage when the main character itself descends down or its vowel identifier is extended down. In such cases the main character itself will be considered as the subscript and hence an anomaly arises.

The proposed VNP algorithm is an improvement over vertical projection profile and it works by segmenting adjacent characters with minimum threshold based on connectedness of the pixels.

METHODOLOGY

System description: The input text images are obtained after scanning printed documents. These documents can either be in grayscale or color but without pictures among the text, like Fig. 2a. The input text image first undergoes binarization. The resulting binary image is then subjected to noise removal. Thinning is then performed to reduce the binary characters into strokes of one pixel width. Then, the lines, words and finally the characters are segmented from the text image.

The architecture of the proposed system is shown in Fig. 1. Binarization, Noise Removal and Thinning are the preprocessing steps for segmentation. The individual stages are explained in the following sections.

Binarization: Images obtained after scanning are generally in either grayscale or color. However, further processing requires the image to be binary. Hence, binarization is performed to convert the input grayscale or color image to its equivalent binary image. Otsu's method is adopted for this purpose. It works by iteratively computing a threshold value to separate the two classes of pixels, black and white, in a way that minimizes the intra class covariance of the pixel values. The threshold is computed using Eq. (1):

$$\sigma^2_b(t) = \sigma^2 - \sigma^2_\omega(t) = \omega_1(t)\,\omega_2(t)[\mu_1(t) - \mu_2(t)]^2 \quad (1)$$

where, weights ω_i are the probabilities of the two classes separated by a threshold t and σ^2_i variances of these classes. The result of binarization is shown in Fig. 2b.

Noise removal: Noise in the image refers to the unwanted pixels or data. There are various factors such as the quality of the paper and the scanner that induce noise into the input image. In this stage we transform the binarized input image to a de-noised image. There exist numerous kinds of noise in images but the most common noise encountered here is pixel noise, which is, the presence of random unwanted pixels across the image. These pixels add extraneous information which affects further processing. For this purpose, a median filter is employed to eliminate such unwanted pixels. The filtered image is shown in Fig. 2c.

Thinning: The filtered image undergoes thinning to obtain the skeleton of the characters in the image without affecting their structural shape. The resulting characters are of one-pixel width and are centered and connected. The results are shown in Fig. 2d.

Line segmentation: The first step in segmentation is line segmentation. Line segmentation, as its name implies, is the separation of the different lines of text. This breaks down the recognition process into the recognition of individual lines. Horizontal projection profile is employed for this purpose. The lines are assumed to be well separated and will hence have separated peaks and valleys in the horizontal projection. The number of black pixels in each row is computed to obtain these peaks and valleys. They are then used to identify the boundary between two lines of text and hence, extract these lines. The result of segmenting the first line from the input image is shown in Fig. 3a.

Word segmentation: After breaking down the input image into lines of text, these lines are broken down into

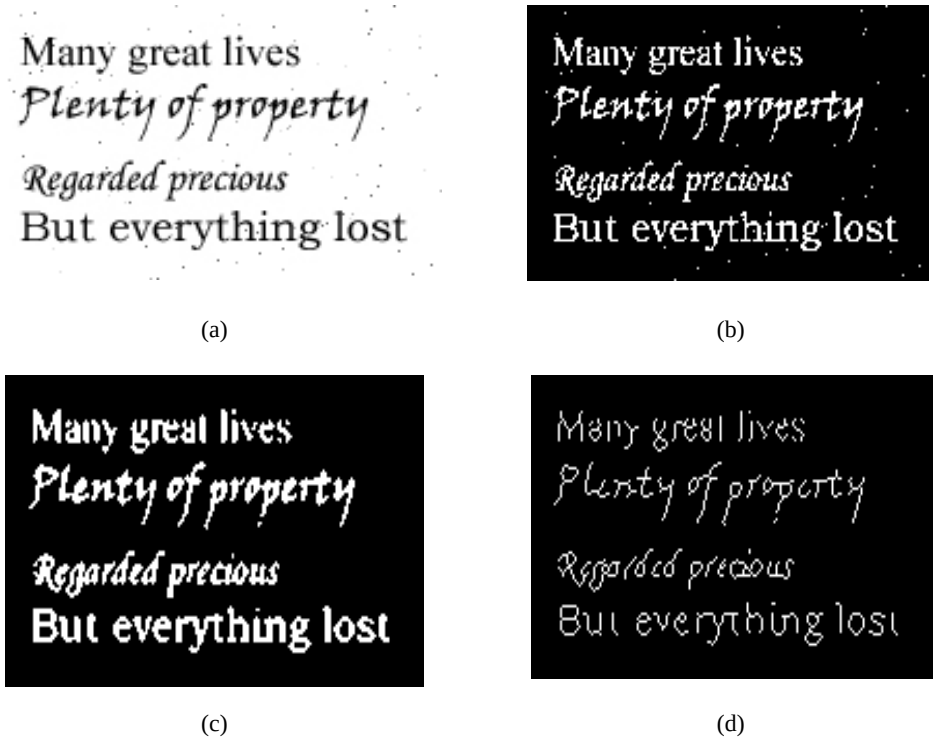

Fig. 2: Preprocessing(a) Input image, (b) Binarized image, (c) Filtered image, (d) Thinned image

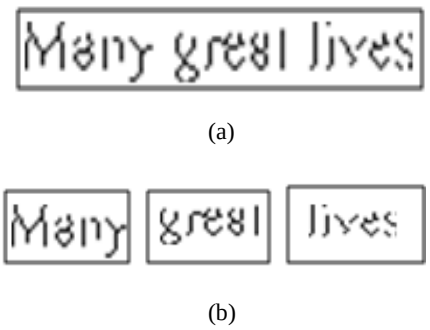

Fig. 3: Segmentation prior to character segmentation, (a) Segmented line, (b) Segmented words

words. Vertical projection profile is used for this purpose. By counting the black pixels along the vertical direction of the line, the valleys of the projection profile are obtained. The gaps in the line can be found by searching for these valleys. These gaps can correspond to either the space between words or the space between characters. Hence, a threshold is set which corresponds to the space between words. Word segmentation is done based on this threshold. The results for word segmentation of the first line are shown in Fig. 3b. The words are boxed to emphasize the segmentation.

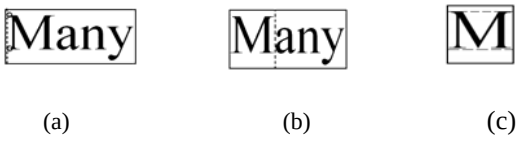

Fig. 4: General character segmentation, (a) Left boundary, (b) Right boundary, (c) Top and bottom boundaries

Character segmentation based on VNP algorithm: Word segmentation is typically followed by character segmentation. Character segmentation is a significant step in many applications. Incorrect segmentation tends to affect the outcome of further processes, like character recognition or classification. An efficient character segmentation technique is the seed to enhance the overall system's performance. The proposed VNP method is classified under dissection based segmentation approaches, wherein the image is split up into meaningful components. In the proposed algorithm the segmented a black pixel is detected. This pixel marks the beginning of the character as depicted in Fig. 4a and it is temporarily stored in an otherwise empty image of the same size and width as the segmented word. The scan progresses iteratively, checking for the presence of black pixels neighboring the already visited black pixels. The end of the character is determined as the column where no more neighboring black pixels exist. This is shown in Fig. 4b.

Fig. 5: Character segmentation for special case-pixels of adjacent characters on same scan line

In some cases, pixels of two adjacent characters fall on the same scan line as shown in Fig. 5. Here, a difficulty arises in identifying the bounds for separating the characters. The VNP algorithm proves to be advantageous in such cases, by checking for the condition of adjacency with the visited pixels. This row *y* If *valu* gives a clear indication of segmentation as shown in Fig. 4. After estimating the start and end point of a character, the columns corresponding to the starting and the ending point are used as reference lines and scanning is done horizontally from the top to the bottom. When a black pixel is encountered, the corresponding scan line is marked as the top boundary of the character and then scanning continues horizontally from the bottom boundary of the word to the top. Every time a pixel is encountered, the presence of its adjacent pixels is checked. If it has adjacent pixels falling outside the right boundary of the character, the pixel is assumed to belong to another character and is ignored. This solves the problem of segmenting character when pixels of adjacent characters fall on the same scan line. The scanning proceeds till a pixel which has either no adjacent pixels or whose adjacent pixels fall within the character's boundary is encountered. The row corresponding to the pixel is marked as the bottom boundary as shown in Fig. 4c. Hence, the boundaries of the character are defined and the character is segmented along them.

Algorithm: Visited neighbor pixel algorithm

Functions:
STORE (x,y):Stores the pixel corresponding to location *(x,y)*
visited (x,y): Checks if the pixel *(x,y)* has been visited
height (w): Returns the height of the word *w*
value (x,y): Returns the value of the pixel (0-black or 1-white)

Input:
w: Segmented word

Output:
x_{min}: Left most point of the character
x_{max}: Right most point of character
y_{min}: Top most point of character
y_{max}: Bottom most point of character

Start
For word w
//Storing first pixel of character
For x = 1
For-Each row y
// black pixel encountered
If value (x,y) == 0 // Start of the character
Then STORE (x,y)
x_{min} = x
End If
End For
End For
For column x = 2: n
For-Each e (x,y) == 0
//Checking for neighbor black pixel
If (!value (x – 1, y)|| ! value(x – 1, y ± 1)||
! value (x, y ± 1))
If (visited (x – 1, y)|| visited (x, y ± 1)) = = 0)
STORE (x, y)
Else
//Absence of neighbor pixels
$x_{max} = x$
End If
End For
End For
// Computing boundaries of bounding box
y_n = height (w)
For-Each row y from column
If (x, y) = = 0 **break**
STORE (x,y)
y_{min} = y
For y = y_n; y>1; y-1
If ((x_{max} + 1, y)≠0)
y_{max} = y
End If
End For
End For
End

RESULTS AND DISCUSSION

A subset of the available True Type Fonts, in which a few pixels of adjacent character's fall on the scan line, were selected for analyzing the performance of the proposed method. The proposed method for character segmentation was applied to input images containing these fonts namely-Monotype Corsiva, Viner Hand ITC, Times New Roman and Bookman Old Style. The results of character segmentation for a sample word from each of the tested fonts are shown in Fig. 6.

The results of this character segmentation were juxtaposed with those by vertical projection profile and their precision, recall and F-scores were computed. The values for precision, recall and F-score were computed using Eq. (2), (3) and (4), respectively:

$$P = \frac{TP}{TP + FP} \quad (2)$$

$$R = \frac{TP}{TP + FN} \quad (3)$$

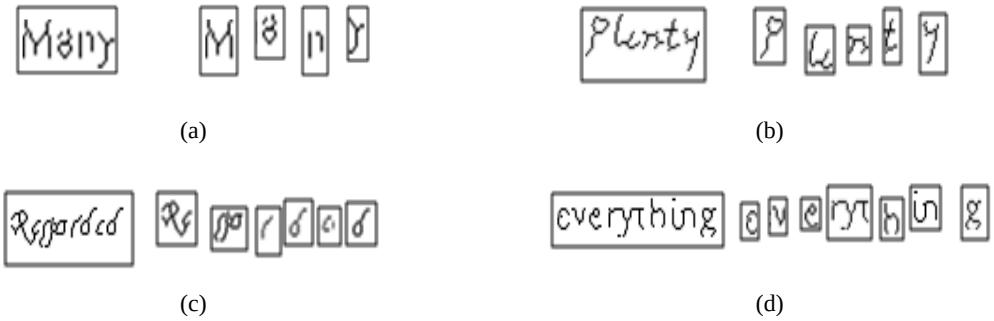

Fig. 6: Character segmentation, (a)Times new roman, (b) Viner hand ITC, (c) Monotype corsiva, (d) Bookman old style

Table 1: Performance comparision of VNP algorithm with vertical projection profile

	VNP algorithm			Vertical projection profile		
Font styles	P	R	F-score	P	R	F-score
Viner hand ITC	0.94	0.96	0.95	0.88	0.95	0.91
Times new roman	0.93	0.94	0.94	0.90	0.96	0.93
Monotype corsiva	0.88	0.95	0.91	0.83	0.90	0.86
Bookman old style	0.94	0.97	0.95	0.92	0.96	0.94

$$F = 2\frac{P.R}{P+R} \quad (4)$$

TP (true positive) pertains to the number of characters segmented correctly. FP (false positive) pertains to the number of characters that remain unsegmented. FN (false negative) pertains to the segmentation results which do not correspond to characters.

The results of comparison of character segmentation using VNP algorithm and that using vertical projection profile are tabulated in Table 1.

In fonts like Viner Hand ITC there are many cases where pixels of adjacent characters fall in the same scan line. Hence, a relatively large difference in precision can be observed when compared to vertical projection profile. The average accuracy for character segmentation, computed as the average of the F-scores, was found to be approximately 94% when using VNP algorithm as opposed to 91% when using vertical projection profile.

The segmentation errors occurring when VNP algorithm is used are primarily attributed to adjacent characters touching each other. However, vertical projection profile also suffers from these errors. This is generally a result of one or more preprocessing steps.

CONCLUSION

The proposed character segmentation technique based on VNP algorithm uses the connectedness of the pixels as a property to distinguish characters. Adjacent characters whose pixels fall on the same scan line are also correctly segmented. The results of the proposed method were compared with those of vertical projection profile. With an average accuracy of about 94%, character segmentation using VNP algorithm showed favorable results.

REFERENCES

Jaedo, K., H. Youngjoon and H. Hernsoo, 2009. Character segmentation method for a license plate with topological transform. World Acad. Sci. Eng. Technolo., 35: 39-42.

Min-Chul, J., S. Yong-Chul and S.N. Sargur, 1999. Machine printed character segmentation method using side profiles. IEEE Int. Conf. Syst. Man Cybernet., 6: 863-867.

Rajiv, S.K. and S. Amardeep, 2008. Segmentation of hand written text in Gurmukhi script. Int. J. Image Proc., 2: 13-17.

Sanjeev, K.R. and S.R.D. Sudhaker, 2006. A two-stage character segmentation technique for printed Kannada text. GVIP Special Issue Image Sampl. Segment., 6: 1-8.

Seong-Whan, L., L. Dong-June and P. Hee-Seon, 1996. A new methodology for gray-scale character segmentation and recognition. IEEE T. Pattern Anal., 18: 1045-1050.