Segmentation of Handwritten Bangla Script

Arafat Rahman, Hossain Md. Cyrus, Farhad Yasir, Waliul Bari Adnan and
Md. Monirul Islam
Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology
Dhaka-1000, Bangladesh

Abstract— **Segmentation of handwritten Bangla script is one of the most critical areas of the Optical Character Recognition System. Paying attention on the various writing style of different individuals we propose an efficient scheme to segment unconstrained handwritten Bangla script into lines, words and characters. At First for Line Segmentation, we divide the whole script into column segment. These segments are calculated by the mode of the width of each black pixel region. In each column segment, we mark potential line markers considering the height of black pixel regions. We compute a set of potential line markers for each segment and join them using the Construct Line Algorithm method. The algorithm is used to segment the text lines. Considering the width of the black pixel regions and computing the distance between two consecutive black pixel regions, lines are segmented into words. In handwritten word, determining the Matra is necessary to segment the word into characters. We take the word into minimum bounding box and consider those black pixels where the vertical flow of white pixels block. The mode of the vertical positions of these black pixels is determined to find the Matra zone where the characters are connected with one another. Considering pixel density of these connections between two characters are determined to divide the words into characters**.

I. INTRODUCTION

Segmentation of handwritten Bangla script is one of the most challenging processes in Bangla Optical Character Recognition System because of slanting style of freestyle handwriting and the overlapping of different character components. Variation of writing style and complex character modifiers of Bangla language complicate the segmentation process hardly. There are few works on Bangla Handwriting Script based on statistical analysis [10]. There are also few proposals on segmentation of overlapping handwritten script. But only one of the proposals is acceptable based on water reservoir principle [2].

In this paper, we propose a statistic-free scheme to segment unconstrained handwritten Bangla script by following computation based techniques. We divide the whole script into column segment pieces which are calculated by the mode of the width of each black pixel region. We introduce potential line markers which mark the row position considering the height and noise removal techniques. Construct Line Algorithm is applied to join the potential line markers of each column segment to divide the script into lines. The approximate width of the black pixel regions and the distance between two consecutive black pixel regions are used to segment the lines into words. Segmentation of character from words is complex process. We compute the Matra zone which is considered as the region where characters are connected with one another. We compute the pixel density of the connections between two characters and divide the word into character on these connections.

II. INTRODUCTION TO BANGLA CHARACTERS

The set of basic characters of Bangla consists of 11 vowels and 39 constants. It contains compound characters and character modifiers which take a major impact on the variation of different word in Bangla language. A consonant or a vowel following a consonant is called as compound character. The following figures show Bangla basic characters, vowel modifiers, and consonant modifiers:

অ	ঊ	ঔ	ঙ	ঞ	ণ	ন	ম	ষ	য়
আ	ঋ	ক	চ	ট	ত	প	য	স	ং
ই	এ	খ	ছ	ঠ	থ	ফ	র	হ	ঃ
ঈ	ঐ	গ	জ	ড	দ	ব	ল	ড়	ঁ
উ	ও	ঘ	ঝ	ঢ	ধ	ভ	শ	ঢ়	ৎ

Figure 1 : Bangla basic characters**.**

া,ি, ী,ু,ূ,ৃ ,ে,ৈ,ো,ৌ

Figure 2 : Vowel character modifiers.

ব্য	ব্র	র্ব

Figure 3 : Consonant character modifiers.

978-1-4799-0400-6/13/$31.00 ©2013 IEEE

III. Proposed Method

A. Pre-Processing

The whole document of Bangla handwriting script is scanned line by line and each pixel of the scanned image is converted to its equivalent binary value. To ensure the white background with black text it may necessary to invert the scanned binary image. Using these techniques, the raw image is prepared for the segmentation process.

B. Line Segmentation

Before line segmentation, it is needed to calculate the approximate width and height of a word from the scanned image. The height also denotes the distance between two consecutive lines. Word of a document represents an approximate area of continuous black pixel.

To determine the approximate area, the whole document is scanned pixel by pixel. If any black pixel appears, consider that pixel as a parent and start visiting its 8 neighbor following Depth-First-Search Algorithm [8]. This search continues until no neighbors has black pixel. After every search a region of black pixels with W_a^i width and H_a^i height is traversed (Figure 4). H_a^i is computed by the distance between the maximum and the minimum vertical positions of black pixels for that region and W_a^i of that region is also computed by the distance between the maximum and the minimum horizontal positions black pixels. For the whole document a number of black pixel regions is traversed and a set of W_a^i and H_a^i is computed by these regions.

$$S_w = \{W_a^i \mid\ Width\ of\ i^{th}\ cluster\}$$

$$S_H = \{H_a^i \mid\ Height\ of\ i^{th}\ cluster\}$$

For better approximation, rather than taking the usual mode of the two sets S_w and S_H, 1st and 2nd mode are taken to determine the approximate width (W_a) and height(H_a).

$$\boldsymbol{W_a} = \frac{\boldsymbol{n_1 * 1^{st}\ mode\ of\ S_w + n_2 * 2^{nd}\ mode\ of\ S_w}}{\boldsymbol{n_1 + n_2}} \qquad \textbf{(1)}$$

$n_1 = No.of\ occurence\ of\ 1^{st}\ Mode\ of\ S_w$

$n_2 = No.of\ occurence\ of\ 2^{nd}\ Mode\ of\ S_w$

$$\boldsymbol{H_a} = \frac{\boldsymbol{n_1 * 1^{st}\ mode\ of\ S_H + n_2 * 2^{nd}\ mode\ of\ S_H}}{\boldsymbol{n_1 + n_2}} \qquad \textbf{(2)}$$

In the above equation n_1 and n_2 is calculated from S_H .

The approximate width of a word W_a is the width of Column Segment (CS).

$$\boldsymbol{No.of\ Column\ Segment(n_{CS})} = \frac{\boldsymbol{W_t}}{\boldsymbol{W_a}} \qquad \textbf{(3)}$$

$W_t = Total\ Width\ of\ the\ sample\ image$

$W_a = Approximate\ Width\ of\ a\ Word$

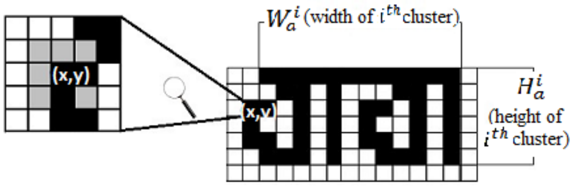
(x,y)
W_a^i (width of i^{th} cluster)
(x,y)
H_a^i
(height of i^{th} cluster)

Figure 4 : Zoom in a pixel level and 8 neighbors of a pixel

For every Column Segment (CS), the sub-image is scanned from top to bottom pixel by pixel to find the Potential Line Markers (*PLM*).Two-steps process is applied on each segment to find the *PLM*s of that segment. In first step, if any black pixel appears, the immediate vertical position above that pixel is marked as a Potential Line Marker (*PLM*).In second step, if any horizontal line of white pixels appears; the immediate vertical position below that line is marked as another *PLM.* These positions are supposed to consider the upper bound and lower bound of a line in that particular CS. This two-step process continues till the end of segment. This approach is used to deal with the random slanting style of handwritten line and determine a set of *PLM*s for every CS. The set of *PLM*s for i^{th} CS is considered as S_{PLM}^i where

$$\boldsymbol{S_{PLM}^i} = \{\boldsymbol{PLM_1}, \boldsymbol{PLM_2}, \boldsymbol{PLM_3} \cdots\cdots \boldsymbol{PLM_n}\} \qquad \textbf{(4)}$$

n = No. of *PLM* for i^{th} Column Segment.

$i = 1,2,3, \cdots\cdots\cdots\cdots, n_{cs.}$

In every S_{PLM}^i there are a few number of *PLM*s that act as noise. These noisy *PLM*s can appear for the following scenarios:

- Detached part above the Matra.
- Detachment of characters from Matra.
- The detached lower section of a character(i.e. the dots below character)
- The character modifiers.

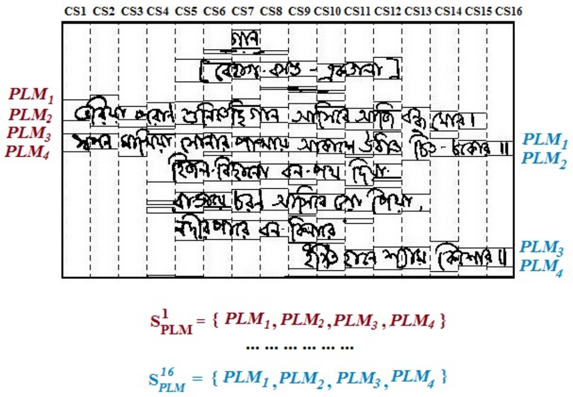
CS1 CS2 CS3 CS4 CS5 CS6 CS7 CS8 CS9 CS10 CS11 CS12 CS13 CS14 CS15 CS16
PLM_1
PLM_2
PLM_3
PLM_4
PLM_1
PLM_2
PLM_3
PLM_4
$S^1_{PLM} = \{PLM_1, PLM_2, PLM_3, PLM_4\}$
… … … … … …
$S^{16}_{PLM} = \{PLM_1, PLM_2, PLM_3, PLM_4\}$

Figure 5 : Initial PLMs.

We devised a robust approach to detect the noisy *PLM* and remove them considering the position of other nearby *PLM*. The approach of removing noisy *PLM*s is given below:

1. Consider the entire PLM_j in S^i_{PLM} where $S^i_{PLM} = \{PLM_1, PLM_2, PLM_3 \cdots\cdots PLM_n\}$.
2. Compute the distance between PLM_j and other PLM_k of this i^{th} $segment$.
3. If the distance is smaller than the approximate height of the word H_a, then determine the distance between PLM_j and its previous *PLM* and the distance between PLM_k and its next *PLM*.
4. If $(PLM_j - PLM_{j-1}) < (PLM_{k+1} - PLM_k)$ than PLM_k is considered as an appropriate marker of the line segmentation. Where as PLM_j is ignored as a noise and removed from the S^i_{PLM} and $S^i_{PLM} = S^i_{PLM} - \{PLM_j\}$.
5. Else PLM_j is considered as an appropriate marker of the line segmentation. Where as PLM_k is ignored as a noise and removed from the S^i_{PLM} *and* $S^i_{PLM} = S^i_{PLM} - \{PLM_k\}$.

For every CS, a set of potential line marker - S^i_{PLM} is computed and these S^i_{PLM} comprise the set S_{PLM} where $S_{PLM} = \{S^1_{PLM}, S^2_{PLM}, S^3_{PLM} \cdots\cdots S^{n_{cs}}_{PLM}\}$.

From the set S_{PLM}, all the adjacent pairs of S_{PLM} (i.e. S^i_{PLM}, S^{i+1}_{PLM}) are taken respectively where the Construct-Line Algorithm is applied to join the markers and construct a structure of line which is used to segment the text into lines. This structure of line sets contains *PLMs* for every segment.

Construct-Line Algorithm

1. Consider each adjacent pairs S^i_{PLM} and S^{i+1}_{PLM} of the set S_{PLM} .
2. All the PLM_j of S^i_{PLM} set are considered to construct the structure of line.
3. If the PLM_j is not a member of any structure of line set L_m , then add a new structure of line set $L_m = \{PLM^i_j\}$.
4. If there is a PLM_k of S^{i+1}_{PLM} set which is not a member of any structure of line set and has the minimum distance from PLM_j, then add this PLM_k to the corresponding structure line set of PLM_j.So L_m will be
$$L_m = L_m \cup \{PLM^{i+1}_k\}$$
where PLM^{i+1}_k is the marker of L_m line set for $i + 1^{th}$ segment.This minimum distance must be smaller than the half of the approximate height of word.
5. If no such PLM_k exists, then add the same PLM_j for the $i + 1^{th}$ segment to the corresponding structure line set of PLM_j.

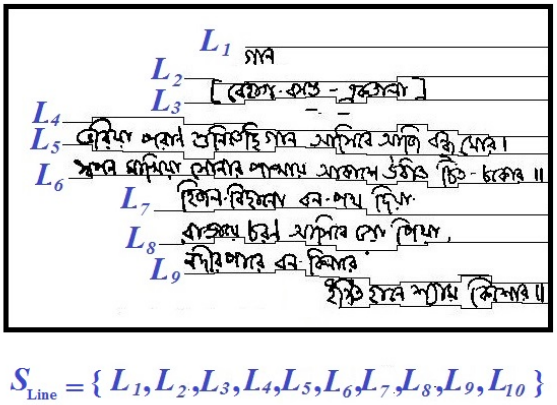
L_1
L_2
L_3
L_4
L_5
L_6
L_7
L_8
L_9
$S_{Line} = \{L_1, L_2, L_3, L_4, L_5, L_6, L_7, L_8, L_9, L_{10}\}$

Figure 6 : Structure Line Set

$S_{Line} = \{L_1, L_2, L_3 \cdots\cdots L_m\}$ where m is the number of line and each member is of following structure. $L_m = \{PLM^i_j, PLM^{i+1}_j, PLM^{i+2}_j, \cdots\cdots, PLM^{n_{cs}}_j\}$ where

PLM^i_j is the j^{th} *PLM* of S^i_{PLM} or i^{th} segment.

For every L_m ,if the number of member PLM^i_j is smaller than the half of the number of Column Segment (CS), that line L_m is considered as noise and remove it from the set of lines S_{Line}.

C. Word Segmentation

A computation based approach is followed to segment the line into words. Each adjacent pairs (L_i, L_{i+1}) of the set $S_{Line} = \{L_1, L_2, L_3 \cdots\cdots L_m\}$ are taken and the white pixel areas which are covered by two consecutive black pixel region are computed. After traversing the whole area between those two adjacent lines, the average width(W_{wa}) of those white pixel areas are computed. To divide the line into word a threshold value ($W_{wa}/2$) is considered as Word Delimiter. Lines are divided in the middle of the Word Delimiters to get the expected words.

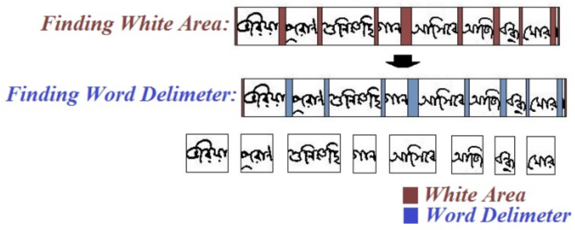

Figure 7 : Word Segmentation

D. Character Segmentation

A Bangla word can be partitioned into three zones. The *upper-zone* denotes the portion with the Matra, the *middle zone* covers the portion between Matra and *lower-zone*, and the *lower-zone* is the portion having character modifiers.

After word segmentation, a set of words(S_{word}) are segmented from a line, each word contains a number of characters with character modifiers. Taking a word from the set - S_{word} and consider a minimum bounding box (h_w*w_W) [9] around that word where h_w is the height and $w_W$is the width of that bounding box.

To find the Matra of a word consider those black pixels where the vertical flow of white pixels block. The mode of the vertical position of these black pixels determine the middle horizon line of a rectangle $({}^{h_w}/_{3}, w_W)$ which denotes the *upper zone or Matra zone.*

Characters in a word are connected with one another through *Matra*. In the *Matra zone,* connections between two characters occur. By scanning the whole *Matra zone* these connections are detected by calculating the pixel density. The minimum pixel density connections of the *Matra zone* are determined to divide the word into Characters.

Following these method a very few character are failed to segment properly (like গ, প, শ etc.)

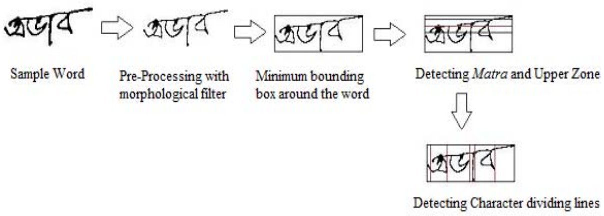
Sample Word
Pre-Processing with morphological filter
Minimum bounding box around the word
Detecting *Matra* and Upper Zone
Detecting Character dividing lines

Figure 8 : Character Segmentation

IV. Experimental Result

The proposed scheme of segmentation is tested mainly on free style Bangla handwritten script image written by different individual. As free style handwriting has slanting style and overlapping that complicate the process to justify the efficiency of that schema. Various handwritten scripts samples are tested in our scheme to gain the proper accuracy. Based on the line word and character segmentation method, our algorithm has gained a high accuracy of almost 96% for line segmentation, 92% for word segmentation, and 64% of character segmentation respectively.

In character segmentation the accuracy level is pretty low because of the limitation of the detection a few character like গ, প, শ etc. These characters are divided into two pieces that cause a bad impact on recognizing the character.

V. Conclusion

In this paper an effective scheme to segment unconstrained handwritten Bangla script is proposed. This proposal is a total statistic free approach where computational based techniques are implemented to segment the Bangla script document into lines, words and characters. We introduce construct line algorithm for the line segmentation and word delimiter to segment the line into words. We have introduced a different approach of the blockage of vertical white pixel flow to determine the *Matra zone* and ease the character segmentation process. In the method of segmenting characters, the accuracy level is not high enough so in future we consider the constraint to make better accuracy on that method.

References

[1] *W. Pan, Xiaojun, DuTien. D. Bui, "Text Line Segmentation in Handwritten Documents, Using Mumford-Shah Model". Pattern recognition, Vol-42, Issues-12, and Dec -2009, Pages: 3136 – 3145.*

[2] *U. Pal and Sagarika Datta, "Segmentation of Bangla Unconstrained Handwritten Text", Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on 6-6 Aug 2003.pages 1128 – 1132.*

[3] *B. B. Chaudhuri* and U. Pal," A complete printed Bangla OCR system". Pattern Recognition, Vol. 31, No. 5, pp. 531-549, 1998.*

[4] *G.Louloudis, B.Gatos, I.Pratikakis, C.Halatsis "Text Line detection in Handwritten Documents". Pattern Recognition 41 (2008) 3758—3772.*

[5] *R. Plamondon, "Online and offline Handwritten Recognition: A comprehensive study" IEE transaction on pattern analysis and machine intelligence. Vol 22, No.1 ,2000*

[6] *S. Basu , Nibaran Das , Ram Sarkar, Mahantapas Kundu, Mita Nasipuri, Dipak Kumar Basu, "A hierarchical approach to recognition of handwritten Bangla characters", Pattern Recognition 42 (2009) 1467—1484.*

[7] *A.F.R. Rahman, R. Rahman, M.C. Fairhurst, "Recognition of handwritten Bengali characters: a novel multistage approach". Pattern Recognition 35 (2002) 997–1006.*

[8] R. G. Casey and E. Lecolinet, "A survey of methods and strategies in character segmentation" IEEE Trans.on PAMI, vol.18, pp. 690 - 706,1996