

# Linear Curve Fitting-Based Headline Estimation in Handwritten Words for Indian Scripts

Rahul Pramanik<sup>(✉)</sup>  and Soumen Bag 

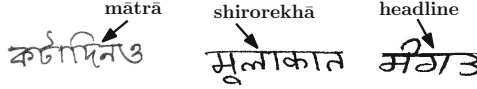
Department of Computer Science and Engineering, Indian Institute of Technology  
(ISM) Dhanbad, Dhanbad, India  
rahul.wbsu@gmail.com, bagsoumen@gmail.com

**Abstract.** Most segmentation algorithms for Indian scripts require some prior knowledge about the structure of a handwritten word to efficiently fragment the word into constituent characters. Zone detection is a considerably used strategy for this purpose. Headline estimation is a salient part of zone detection. In the present work, we propose a method that uses simple linear regression for estimating headlines present in handwritten words. This method efficiently detects headline in three Indian scripts, namely Bangla, Devanagari, and Gurmukhi. The proposed method is able to detect headlines in skewed word images and provides accurate result even when the headline is discontinuous or mostly absent. We have compared our method with a recent work to show the efficacy of our proposed methodology.

**Keywords:** Handwritten words · Headline estimation · Indian scripts · Linear regression

## 1 Introduction

Segmentation is one of the most consequential phase in optical character recognition. Presence of cursiveness in Indian scripts makes the segmentation task much more harder [1]. Most segmentation algorithms for Indian scripts require some prior knowledge about the structure of a handwritten word to efficiently fragment the word into constituent characters. Zone detection is a considerably-used strategy for this purpose. Zone detection separates a word into three segments, namely upper, middle, and lower zone. The upper zone is detected by exploiting the headline, a special feature present in most Indian scripts. Sarkar *et al.* [2] have computed the headline in Bangla words by extracting horizontalness and verticalness features from the words. Roy *et al.* [3] have estimated headline in Bangla words using the height of the word, horizontal projection analysis and certain heuristics. Bag and Krishna [4] have used horizontal density row and local maximum row for detecting headlines in handwritten Hindi words. But, these methods suffer when the words are skewed or when the headline is discontinuous or mostly absent. Furthermore, there is an inadequacy of methodologies that are capable of handling multi-script in a document.



**Fig. 1.** Horizontal line called *mātrā* in Bangla, *shirorekhā* in Devanagari script, and headline in Gurmukhi script.

In the present work, we propose a method that uses simple linear regression for estimating headlines present in handwritten words. This method efficiently detects headline in three Indian scripts, namely Bangla, Devanagari, and Gurmukhi. It can be effectively used in word images extracted from a document comprising of multi headline-based script without any prior knowledge about the scripts. The proposed method is able to detect headlines in skewed word images and provides accurate result even when the headline is discontinuous or mostly absent.

The rest of the paper is organized as follows. The proposed methodology is delineated in Sect. 2. In Sect. 3, the experimental results and analysis are discussed, followed by conclusion in Sect. 4.

## 2 Proposed Method

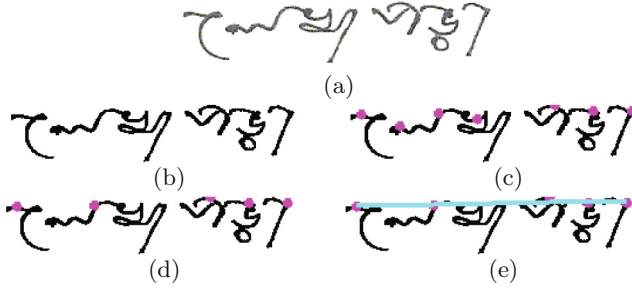
Most Indian scripts have a distinctive feature called headline (also known as *mātrā* in Bangla and *shirorekhā* in Devanagari script) (Fig. 1) present in words. All the characters are connected by the headline at the upper portion of a word. This headline is sometimes discontinuous depending on the individuality of a person’s handwriting. We propose a strategy that utilises this distinctive feature to estimate headline present in word images. The method employed is very simple and effective.

### 2.1 Preprocessing

Initially we binarize all gray level word image ( $\tau_k$ ) (Fig. 2a). We denote the binarized image as  $\nu(\tau_k)$ . We use Rosenfeld and Kak component labelling algorithm to label all the connected components and calculate the size (w.r.t. total number of pixels) of each connected component in  $\nu(\tau_k)$ . Next, we remove each connected component that appears on the top three-fourth of  $\nu(\tau_k)$  and constitutes pixels below a certain threshold,  $\rho(=30)$  as a noise normalization procedure. We have used 150 word images for the validation of the optimal value of  $\rho$ .

### 2.2 Headline Estimation

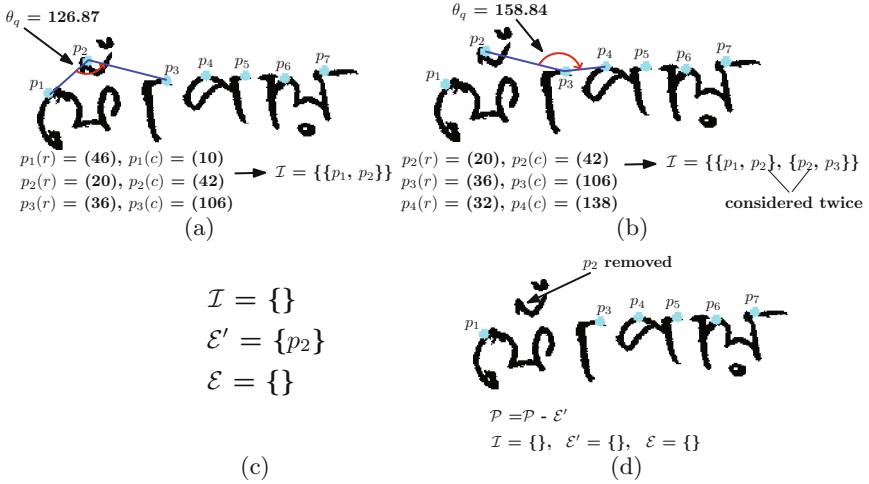
We take  $\nu(\tau_k)$  with dimension  $m \times n$  as input (Fig. 2b) and select  $q$  of the  $n$  columns in  $\nu(\tau_k)$  based on a predefined distance. These  $q$  selected columns are denoted as  $\mathcal{Q} = \langle c_1, c_2, \dots, c_q \rangle$ . The columns are selected at a distance



**Fig. 2.** Stepwise illustration of headline estimation. (a) Input grayscale image; (b) After binarization and noise normalization; (c) First encountered foreground pixels on equidistant columns marked with magenta colour; (d) After removal of ineligible pixels; (e) Estimated headline marked with cyan colour.

of 12% of the width of  $\nu(\tau_k)$ . We used a subset of 350 images to validate this optimal percentage. We detect and store the first encountered foreground pixel on each column in  $\mathcal{Q}$  while traversing  $\nu(\tau_k)$  from top to bottom. The stored foreground pixels are denoted as  $\mathcal{P} = \langle p_1, p_2, \dots, p_q \rangle$  (Fig. 2c). Each stored pixel  $p_i$  is associated with a row and a column number denoted as  $p_i(r)$  and  $p_i(c)$  respectively. We use three sets namely,  $\mathcal{E}$ ,  $\mathcal{E}'$ , and  $\mathcal{I}$  to represent the pixels in  $\mathcal{P}$  as eligible, ineligible, and intermediate respectively.  $\mathcal{E}$  represents pixels that are eligible for further headline estimation.  $\mathcal{E}'$  represents pixels that will be deleted from  $\mathcal{P}$  and will not be considered for further computation.  $\mathcal{I}$  represents pairs of pixels that are temporarily put here before checking their belongingness in  $\mathcal{E}$  or  $\mathcal{E}'$ . For every three consecutive pixels  $p_i$ ,  $p_{i+1}$ , and  $p_{i+2}$  in  $\mathcal{P}$ , we evaluate the angle  $\angle p_i p_{i+1} p_{i+2}$  (denoted as  $\theta_q$ ). If  $\theta_q \leq 165^\circ$ , we conclude that any one of the three pixels is not a headline pixel. To determine which pixel among the three is not a headline pixel, we compute  $|p_i(r) - p_{i+1}(r)|$  and  $|p_{i+1}(r) - p_{i+2}(r)|$ . The difference of column values of the pixels in  $\mathcal{P}$  will mostly be zero, as the columns are equi-distant from each other. So, we only take row values of pixels in  $\mathcal{P}$  into consideration for distance computation. We have used a subset of 150 word images to validate 165 as the optimal angle. If  $|p_i(r) - p_{i+1}(r)| > |p_{i+1}(r) - p_{i+2}(r)|$ , then we conclude that either  $p_i$  or  $p_{i+1}$  is not a headline pixel and as a result we consider  $\{p_i, p_{i+1}\}$  as intermediate pixel pair and store the pair in  $\mathcal{I}$ . Otherwise, we conclude that either  $p_{i+1}$  or  $p_{i+2}$  is not a headline pixel and consider  $\{p_{i+1}, p_{i+2}\}$  as intermediate pixel pair and store the pair in  $\mathcal{I}$  instead. If a pixel  $p_i$  in  $\mathcal{P}$  is considered twice as intermediate in a single iteration, then we conclude that  $p_i$  is not a headline pixel and transfer  $p_i$  from the set  $\mathcal{I}$  to  $\mathcal{E}'$ , while the pixel paired with  $p_i$  in  $\mathcal{I}$  is removed from  $\mathcal{I}$ . Once all the intermediate pixels are marked in a single iteration, we compute the eligible pixels in  $\mathcal{E}$  as  $\mathcal{E} = \mathcal{P} - (\mathcal{I} \cup \mathcal{E}')$ .

For every pixel pair  $\{p_i, p_{i+1}\}$  in  $\mathcal{I}$ , we compute the row-wise difference,  $df_{p_i}$  and  $df_{p_{i+1}}$  of  $p_i$  and  $p_{i+1}$  with every pixel in  $\mathcal{E}$ . We compute the maximum of the two differences  $df_{p_i}$  and  $df_{p_{i+1}}$  as  $max_{df}$ . A non-headline pixel will always



**Fig. 3.** Stepwise illustration of ineligible pixel removal. (a) Angle  $\angle p_1 p_2 p_3 \leq 165^\circ$  and  $|p_1(r) - p_2(r)| > |p_2(r) - p_3(r)|$ , so,  $\{p_1, p_2\}$  is stored in  $\mathcal{I}$ ; (b) For the next three pixels, angle  $\angle p_2 p_3 p_4 \leq 165^\circ$  and  $|p_2(r) - p_3(r)| > |p_3(r) - p_4(r)|$ , so,  $\{p_2, p_3\}$  is stored in  $\mathcal{I}$ , but two consecutive pixel pairs contain the same pixel, i.e.,  $p_2$ ; (c) As  $p_2$  consecutively appears twice in  $\mathcal{I}$ , so  $p_2$  is transferred to  $\mathcal{E}'$ , while the two pixels associated with it, i.e.,  $p_1$  and  $p_3$ , are removed; (d) Pixels in  $\mathcal{E}'$  are removed from  $\mathcal{P}$ .

have a greater row difference with headline pixels than the difference between a headline pixel with other headline pixels. So, the pixel in the pixel pair  $\{p_i, p_{i+1}\}$  that is associated with most number of  $max_{df}$  is transferred from  $\mathcal{I}$  to  $\mathcal{E}'$  while the other is transferred to  $\mathcal{E}$ . Once all the pixel pairs in  $\mathcal{I}$  are checked, the pixels belonging to  $\mathcal{E}'$  are removed from  $\mathcal{P}$  (Fig. 2d).  $\mathcal{E}$ ,  $\mathcal{I}$ , and  $\mathcal{E}'$  are all emptied. This procedure is carried out until no three consecutive pixel in  $\mathcal{P}$  creates an angle less than or equal to  $165^\circ$ . We remove ineligible pixels from  $\mathcal{P}$  to ensure that the headline estimation does not get affected due to the presence of upper modifiers and certain consonants that appear above the headline in a word.  $\hat{\imath}$ ,  $\hat{\lrcorner}$ ,  $\hat{\lrcorner}$ ,  $\hat{\lrcorner}$  and  $\hat{\smile}$  are some examples of upper modifier and consonant that appear above the headline in Bangla script. Example of headline estimation of words with such modifiers appearing in Indian scripts are shown in the next section.

We use the word  $\hat{\smile}\hat{\lrcorner}\hat{\lrcorner}\hat{\lrcorner}$  as a working example to demonstrate the proposed methodology. Due to the presence of the consonant  $\hat{\smile}$ , the second pixel  $p_2$  in  $\mathcal{P}$  is marked much higher compared to the position of headline (Fig. 3a). As a result, for the first three pixels in  $\mathcal{P}$ , angle  $\angle p_1 p_2 p_3$  is  $\leq 165^\circ$  and  $|p_1(r) - p_2(r)| > |p_2(r) - p_3(r)|$ . We conclude that either  $p_1$  or  $p_2$  is a non-headline pixel and store  $\{p_1, p_2\}$  in  $\mathcal{I}$  as a pixel pair (Fig. 3a). Again, when we shift one pixel right and consider the next three pixels, angle  $\angle p_2 p_3 p_4$  becomes  $\leq 165^\circ$  and  $|p_2(r) - p_3(r)| > |p_3(r) - p_4(r)|$ . So, we conclude that either  $p_2$  or  $p_3$  is a non-headline pixel and we store  $\{p_2, p_3\}$  in  $\mathcal{I}$  (Fig. 3b). As,  $p_2$  repeats in two

consecutive pixel pairs in  $\mathcal{I}$ , we infer that  $p_2$  is a non-headline pixel. As a result, we transfer  $p_2$  from  $\mathcal{I}$  to  $\mathcal{E}'$  and remove the pixels associated with  $p_2$ , i.e.,  $p_1$  and  $p_3$ , from  $\mathcal{I}$  (Fig. 3c). We check all the remaining consecutive pixels in  $\mathcal{P}$ . Once checking completes, we remove the ineligible pixels in  $\mathcal{E}'$  from  $\mathcal{P}$  (Fig. 3d).

Now, we predict the row values  $\mathcal{P}(\hat{r})$  based on the row and column values of pixels in  $\mathcal{P}$  using the following equation:

$$\mathcal{P}(\hat{r}) = b_0 + b_1 \times \mathcal{P}(c) \quad (1)$$

where,

$$b_1 = \sum_{i=1}^{|\mathcal{P}|} \frac{(p_i(c) - \overline{p(c)})(p_i(r) - \overline{p(r)})}{(p_i(c) - \overline{p(c)})^2},$$

$$b_0 = \overline{p(r)} - b_1 \times \overline{p(c)},$$

$$\overline{p(c)} = \frac{\sum_{i=1}^{|\mathcal{P}|} p_i(c)}{|\mathcal{P}|} \quad \text{and} \quad \overline{p(r)} = \frac{\sum_{i=1}^{|\mathcal{P}|} p_i(r)}{|\mathcal{P}|}.$$

We use the polyfit function in Matlab to employ these equations. Based on the  $\mathcal{P}(\hat{r})$  and  $\mathcal{P}(c)$  values, we draw a regression line which gives the final estimated headline of each word (Fig. 2e).

### 3 Experimental Results and Analysis

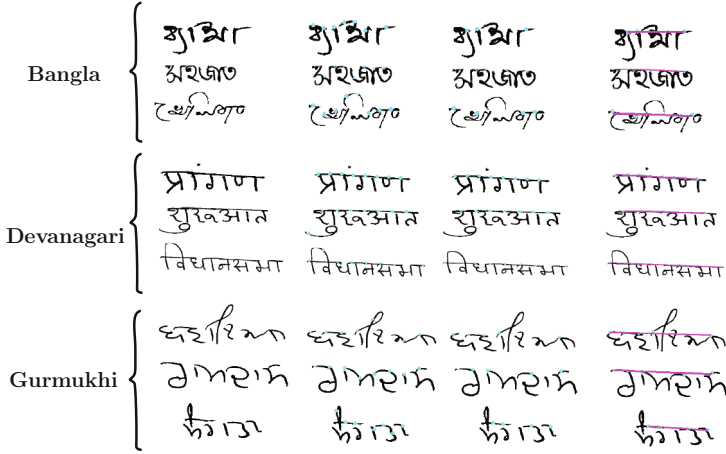
#### 3.1 Dataset

For experimentation, we have used four datasets for three different scripts, namely Bangla, Devanagari, and Gurmukhi. For Bangla script, we have used Cmaterdb dataset version 1.1.1 [5] and ICDAR 2013 Segmentation Dataset [6]. For Devanagari and Gurmukhi script, we have used Cmaterdb dataset version 1.5.1 [7] and PHDIndic\_11 [8] dataset respectively. A total of 4050 words are used for our current experimentation. We have used Matlab for the entire implementation part.

#### 3.2 Test Results and Comparative Analysis

We have delineated the experimental results and analysis of our proposed work in this section. Few outputs of our proposed technique are shown in Fig. 4. Last two rows of each script in Fig. 4 delineates the removal of ineligible pixels due to the presence of upper modifiers as discussed in previous section. A detailed analysis of the headline estimation performance achieved in each script is provided in Table 1. As per the tabulated results, Devanagari and Gurmukhi script provides the most and least precise result with an accuracy of 96.15% and 89.41% respectively. We achieved an overall accuracy of 92.59% when accuracy of all the 3 scripts are considered.

The efficiency of our proposed method is compared with Sarkar *et al.* [2]. This method utilises sum of length of horizontal runs, maximum horizontalness,



**Fig. 4.** Test results on different Indian scripts. **First column:** Word images; **Second column:** First encountered foreground pixels on equi-distant columns marked with cyan colour; **Third column:** Eligible pixels are kept while ineligible pixels are discarded; **Fourth column:** Estimated headline marked with magenta colour.

**Table 1.** Headline estimation accuracy achieved in different Indian scripts.

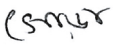
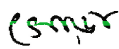
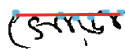
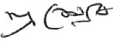


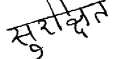

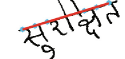



Script	Total # words	# Words with correct headline estimation	Accuracy (%)
Bangla	1350	1245	92.22
Devanagari	1350	1298	96.15
Gurmukhi	1350	1207	89.41
Overall	4050	3750	92.59

**Table 2.** Comparison of our proposed method with Sarkar *et al.* [2].

Method	Script	Total # words	# Words with correct headline estimation	Accuracy (%)
Sarkar <i>et al.</i> [2]	Bangla	500	397	79.40
Proposed method			467	93.40

horizontalness, and verticalness feature to identify the headline in handwritten Bangla words before segmentation is performed. This method is limited to handle non-skew words and also provides inaccurate result when the headline is mostly absent. Our proposed method is able to provide accurate result even when the headline is mostly absent and can handle skewed word images as well. A comparison of our proposed method with [2] has been provided in Table 2. We have also provided a visual comparison of few word images with [2] in Table 3

**Table 3.** Headline estimation comparison of few word images of our proposed method with Sarkar *et al.* [2]. Input for the last row is a synthetically oriented word image at  $30^\circ$ .

Original Image	Headline Estimated by [2]	Headline Estimated by our proposed method
		
		
		
		

demonstrating that our proposed method provides more accurate headline estimation than [2].

## 4 Conclusion

Most segmentation algorithms require some prior knowledge about the location of the headline to swiftly and efficiently fragment a handwritten word into constituent characters in majority of Indian scripts. In the present work, we have proposed a method that uses simple linear regression for estimating headline present in handwritten words. This method efficiently detects headline in three Indian scripts, namely Bangla, Devanagari, and Gurmukhi. The proposed method is able to detect headlines in skewed word images and provides accurate result even when the headline is discontinuous or mostly absent.

## References

1. Bag, S., Harit, G.: A survey on optical character recognition for Bangla and Devanagari scripts. *Sadhana* **38**(1), 133–168 (2013)
2. Sarkar, R., Das, N., Basu, S., Kundu, M., Nasipuri, M., Basu, D.K.: A two-stage approach for Segmentation of Handwritten Bangla word Images. In: *Proceedings of ICFHR*, pp. 403–408 (2008)
3. Roy, P.P., Dey, P., Roy, S., Pal, U., Kimura, F.: A novel approach of Bangla handwritten text recognition using HMM. In: *Proceedings of ICFHR*, pp. 661–666 (2014)
4. Bag, S., Krishna, A.: Character segmentation of hindi unconstrained handwritten words. In: Barneva, R.P., Bhattacharya, B.B., Brimkov, V.E. (eds.) *IWCIA 2015*. LNCS, vol. 9448, pp. 247–260. Springer, Cham (2015). doi:[10.1007/978-3-319-26145-4\\_18](https://doi.org/10.1007/978-3-319-26145-4_18)
5. Sarkar, R., Das, N., Basu, S., Kundu, M., Nasipuri, M., Basu, D.K.: CMATERdb1: a database of unconstrained handwritten Bangla and BanglaEnglish mixed script document image. *IJDAR* **15**(1), 71–83 (2012). Accessed 8 Feb 2017
6. Stamatopoulos, N., Gatos, B., Louloudis, G., Pal, U., Alaei, A.: ICDAR 2013 handwriting segmentation contest. In: *Proceedings of ICDAR*, pp. 1402–1406 (2013). Accessed 12 Mar 2017

7. CMATERdb 1.5.1: <http://archive.is/xDqG6#selection-621.0-623.41>. Accessed 2 Jan 2017
8. Das, N., Halder, C., Obaidullah, S.M., Roy, K., Santosh, K.C.: PHDIndic\_11: page-level handwritten document image dataset of 11 official Indic scripts for script identification. In: Multimedia Tools and Applications, pp. 1–36 (2017)