# A New Approach for line segmentation in Punjabi Language using Strip based Projection profile method

Rahul Garg
*GZS-PTU Campus, Bathinda*
rg.rahul91@gmail.com

Naresh Kumar Garg
*GZS-PTU Campus, Bathinda*
naresh2834@rediffmail.com

**Abstract-** *In this paper, a new algorithm has been proposed that can perform line segmentation in handwritten text. This algorithm is based on projection profile technique. The projection profile is then used to find the gap between two text lines. The algorithm mainly deals with skewed text. To deal with touching and overlapping of text lines, strip based technique is used. The algorithm has given comparable results with the algorithms present in literature. Comparison of results of the algorithm with the other algorithms present in literature is presented in this paper.*

**Keywords-** *Text Line segmentation, Projection profile, Gurumukhi language, handwritten text, overlapping, skew, text blocks.*

## I. INTRODUCTION

Handwritten Hindi text recognition is an important area of Optical Character Recognition (OCR) [1]. Improper line segmentation leads to wrong results in word or character segmentation and in recognition. Hence before these, proper line segmentation must be done. Text line segmentation in machine printed text is much easier than handwritten text and it is considered to be a solve problem as in printed text, there is a equal line spacing and there is a same font size in the text. But on the other hand, in handwritten text, usually there is no equal line spacing among the text lines and also there may be overlapping and touching of lines which makes line segmentation a challenging task. Usually the handwritten text is skewed, means text is aligned to some angle which also makes line segmentation difficult. In Punjabi language, there are many upper and lower modifiers and many other diacritics which makes line segmentation very complex as it is not a easy task to segment these. Gurumukhi script is used to write Punjabi language. There is rich literature in Punjabi language in the form of scripture, books, poetry. Very less research work has been done on Punjabi language and very few research papers are available. It is, therefore, important to develop offline handwriting recognition for such a rich and widely used language which may find many practical uses in various areas. There are various other methods line segmentation methods are reported in literature. The various existing methods can be categorised as Projection profile based [2,3], Smearing method [4], Hough transform based [5], Graph based [6]. In this paper we have proposed a new algorithm that can perform line segmentation in handwritten text. The algorithm is based on projection profile technique. Projection profile methods are based on top-down algorithms which are one of the most successful methods in machine printed text [1], but we have efficiently used it in handwritten text. Here vertical projections are obtained by summing the pixel values along the horizontal axis for each value of y. The projection profile is then used to find the gap between two text lines. The algorithm has been applied on many of the handwritten text document images and satisfactory results have been achieved.

## II. DATABASE

All the experiments are have been conducted on database constructed by taking handwritten data from 20 different writers. All writers were asked to write 5-6 lines of some Punjabi. Data of different slant and size is also included. Figure 1 contains part of database. No pre-processing is done on data.

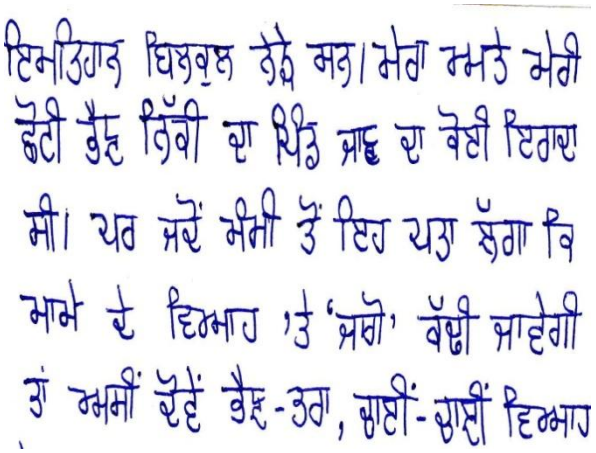

Figure 1. Part of database

## III. CHARACTERISTICS OF PUNJABI LANGUAGE

Gurumukhi script is used to write Punjabi language. The writing style of Gurumukhi is from left to right. There are 41 consonants and 12 vowels in Gurumukhi script. There is no concept of lower or upper case characters in Gurumukhi script. In Gurumukhi script, most of characters have a horizontal line at the upper part. Gurumukhi is a two dimensional composition of symbols with connected and disconnected diacritics[7]. Mostly overlapping or touching of lines occurs due to these diacritics. In most cases, these diacritics are wrongly separated specially the disconnected ones.

## IV. SEGMENTATION

We have proposed an algorithm for handwritten text line segmentation. This algorithm is based on projection profile technique. Line segmentation is done by using the gap between the text lines. To deal with overlapping or touching of lines, the whole document image is divided into 6,7 and 8 equal strips and results have been calculated on many different document images. This algorithm efficiently deal with skewed text but it can also given promising results with overlapped and touched text lines. The algorithm has been discussed in [8]. Algorithm with little modification is explained below.

We have made following assumptions about the data:

1) The minimum height of a constant in a text line is 8 pixels.

2) The average height (AVGHYT) of a text line is between 20 to 40 pixels.

3) The maximum height of a text line (consonant + modifier) is 25 pixels.

4) If a text block is of less than 8 pixels, it would be merged with previous or next text block depending upon condition.

5) If a text block is of more size than AVGHYT+8 pixels, it must be broken into two parts.

Procedure for text line segmentation:

We have initialized the following arrays:

- m_MyImage: It contains pre-processed binarized image.
- HProfiles: It is array that stores the total number of black pixels per row.
- START: It is a 2-D array and contains the starting address of each text block in each strip.
- END: It is a 2-D array and contains ending address of each text block in each strip.

Step-1: Read the input image and then binarize it and store it in a 2-D array m_MyImage.

Step-2: Get information about the height (h) and width of the image (w) which is the size of the 2-D array.

Step-3: Divide the image into vertical strips and for each strip, calculate the number of black pixels in each row and store the result in HProfiles[y] array.

Step-4: For each strip follow these steps:

i. if HProfiles[y]<=0 or HProfiles[y]==1 or HProfiles[y]==2, assign RED color to that

pixel and assign that value of HProfiles[y] to a new variable p.

ii. if HProfiles[y]>p, put the starting pixel address into START array, now a while loop is called that processes all the pixels till the HProfiles[y]==0. Put the last pixel address into END array .

iii. Now we have got the starting and ending addresses of all the text blocks. By using a loop, display the first text block of all the stipes and then after some gap display second text blocks.

Step-5: Repeat the step-4 until full text document image is displayed.

During segmentation with this algorithm, we face the following problems

*Overlapping and Touching of Characters:*

Due to overlapping or touching of characters, there remains no significant gap between the text lines and hence two or more text lines comes in a same text block which leads to wrong results. Hence this bigger text block must be divided into parts at a point where the pixel density is minimum. Figure 2 shows overlapping and touching components.

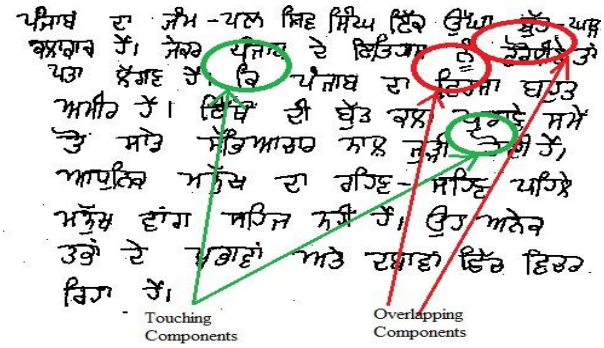


Figure 2. Overlapping and touching components in text lines.

***Disconnected Diacritics:***

There are many disconnected diacritics in Punjabi language that are not connected to the character as shown in figure 3. Hence there got separated in smaller blocks from the text line during line segmentation as shown in figure 4.

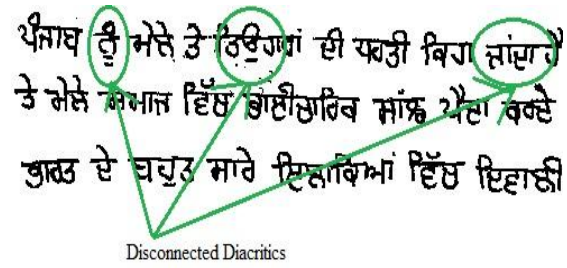


Figure 3. Disconnected Diacritics.

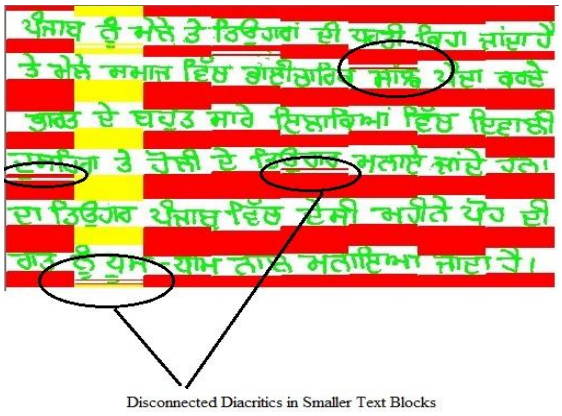


Figure 4. Smaller Text Blocks (in black circles).

To deal with these problems, following steps are added in above algorithm.

Step-1: Calculate the average height of first strip.

Size of text block = END[0][0]-START[0][0].

Total size of all text blocks in a strip, s=s+END[0][j]-START[0][i].

where 0 represnts first strip and i,j= 0,1.. upto 1 less than number of text lines. Initially S=0.

Average height (AVGHYT)= s/n, where n= number of text blocks.

Step-2: Calculate the height(HYT) of each text block.

Step-3: if HYT<8 pixels, it means the height of block is very less means either it is a diacritics or broken character due to poor scanning and hence it must be merged with the previous or next block.

- if it is a lower diacritics then it is merged with pervious block.
- if it is a upper diacritics then it is merged with next block.
- if it is a broken character, then also it is merged with previous block.

For the merging of blocks, it must be identified that whether the block is upper block or lower block so that it could be processed accordingly.

- For this prediction, we have made following assumption.
- We have assumed that the height of a diacritics is 8 pixels.
- Now after the height of diacritics means after 8th pixel, some pixels would be blank. Atleast there would be a gap of 5-8 pixels.
- So we have checked for the 13th pixel (8+5) from the starting of diacritics, if the pixel is blank, then there must be large gap of pixels between the current block and next block hence it must be a lower text block and hence must be attached to the previous block.
- If after the height of diacritics means after 8th pixel, 5th pixel is not blank, then there must be some pixels means there is very small gap between the current block and next block , so it must be a upper block and hence must be attached to the next block.
- So according to these conditions, the small text blocks can be processed.

Step-4: if HYT>= (AVGHYT+8), it means this text block has a very large height hence there must be a touching or overlapping characters.

Here then minima (MIN) is calculated means a point where pixel density is minimum. A while loop is called which processes all the pixels till HProfile[y]==MIN. At a point of MIN, the text block is divided into 2 parts. Starting address of block is stored in START and ending address is stored in END array.

## V. RESULTS

The results of line segmentation are given in the following table. For validation purpose, we have applied the algorithm on Punjabi language text document images and results have been calculated which are also given in following table.

TABLE 4.1: Accuracy of segmentation by dividing Punjabi text document image into 6 strips.

| Total Lines | Lines Correctly Segmented | Percentage of Accuracy |
|---|---|---|
| 120 | 115 | 95.8 |

TABLE 4.2: Accuracy of segmentation by dividing text document image into 7 strips.

| Total Lines | Lines Correctly Segmented | Percentage of Accuracy |
|---|---|---|
| 120 | 106 | 88.4 |

TABLE 4.3: Accuracy of segmentation by dividing text document image into 8 strips.

| Total Lines | Lines Correctly Segmented | Percentage of Accuracy |
|---|---|---|
| 120 | 110 | 91.6 |

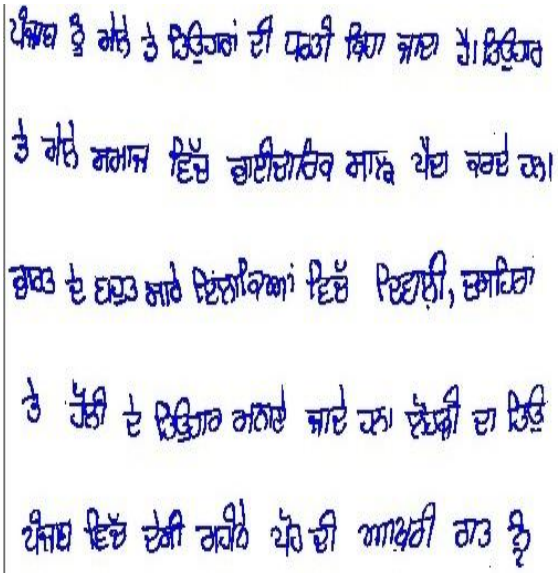

Figure 5. Correctly Segmented Lines.

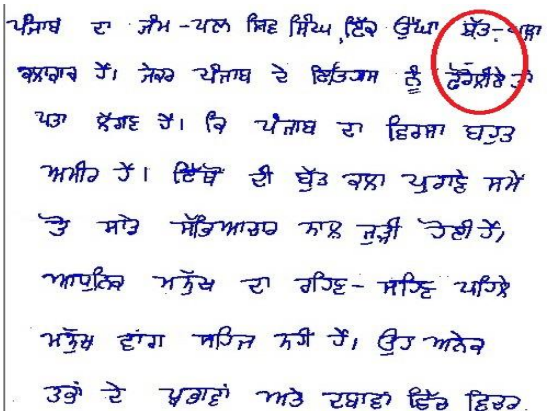

Figure 6. Wrongly Segmented Lines.

## VI. DISCUSSIONS

From the above results, it is clear that the technique given for line segmentation in handwritten Punjabi text has given satisfactory. It has worked well for skewed text as shown in figure 5. The algorithm has resolved the problem of disconnected diacritics. Some problem occurs in presence of overlapping or touching of lines as shown in figure 6. In presence of these, results are not very good but are satisfactory. Our algorithm has given better results with the other algorithms in literature.

| Name of Author | Method Used | Percentage of Accuracy |
|---|---|---|
| M. Kumar et. al. | Strip Based Projection Profile | 93.7 |
| N. Modi, K.Jindal | Projection Profile | 75.78 |
| R. Garg, N. K. Garg | Strip Based Projection Profile with Text Block Merging | 95.8 |

In future the study may be carried out with the following directions:

1) Implementing this algorithm for other languages.

2) Some other method can be implemented for overlapped and touched text line segmentation which can give more accurate results.

## REFERENCES

[1] N. K. Garg, L. Kaur and M. K. Jindal, "A New Method for Line Segmentation of Handwritten Hindi Text", In: Proceedings of the 7th Internation IEEE Conference on Human Technology: New Generations (ITNG), pp. 392-397, 2010.

[2] Bar-Yosef, N. Hagbi, K. Kedem, I. Dinstein, "Line segmentation for degraded handwritten historical documents", International Journal of Document Analysis and Recognition. Vol. 10, pp. 1161-1165, 2009.

[3] Y. Gao, X. Ding, C. Liu, " A Multi-scale Text Line Segmentation Method in Freestyle Handwritten Documents", International Conference on Document Analysis and Recognition, pp. 643-647, 2011.

[4] Z. Shi, and V. Govindaraju, "Line separation for complex document images using fuzzy runlength," First International Workshop on Document Image Analysis for Libraries, pp. 306-312, 2004.

[5] G. louloudis, B. Gatos, I. Pratikakis, C.Halatsis, "Text Line Detection in handwritten documents," Pattern Recognition Vol. 41, pp. 3758-3772, 2008.

[6] I.S.I. Abuhaiba, S. Datta and M. J. J. Holt, "Line Extraction and Stroke Ordering of Text Pages", in the Proceedings of Third International Conference on Document Analysis and Recognition, Montreal, Canada, pp. 390-393, 1995.

[7] N. Modi and K. Jindal, "Text line Detection and Segmentation in Handwritten Gurumukhi Scripts," International Journal of Advance Research in Computer Science and Software Engineering, Vol. 3, pp. 1075-1080, 2013.

[8] R. Garg and N. K. Garg, "An algorithm for Text Line Segmentation in Handwritten Skewed and Overlapped Devanagari Script" International Journal of Emerging Technology and Advanced Engineering, Vol. 4, issue 5, pp. 114-118, 2014.

[9] M. Kumar, R. K. Sharma and M. K. Jindal, "Segmentation of Lines and Words in Handwritten Gurmukhi Script Documents", in Proceedings of first International Conference on Intelligent Interactive Technologies & Multimedia (IITM 2010), pp. 27-30, 2010.