The Segmentation of Half Characters in Handwritten Hindi Text

Naresh Kumar Garg[1], Lakhwinder Kaur[2], and M.K. Jindal[3]

[1] GZS Collage of Engineering & Tech. Bathinda, Punjab, India
naresh2834@rediffmail.com
[2] Dept. of Computer Engineering, UCOE, Punjabi University, Patiala, Punjab, India
mahal2k8@yahoo.com
[3] Panjab University Regional Centre, Muktsar, Punjab, India
manishphd@rediffmail.com

Abstract. Character recognition is an important stage of any text recognition system. In Optical Character Recognition (OCR) system, the presence of half characters decreases the recognition rate. Due to touching of half character with full characters, the determination of presence of half character is very challenging task. In this paper, we have proposed new algorithm based on structural properties of text to segment the half characters in handwritten Hindi text. The results are shown for both handwritten Hindi text as well as for printed Hindi text. The proposed algorithm achieves the segmentation accuracy as 83.02% for half characters in handwritten text and 87.5% in printed text.

Keywords: Segmentation, half character, over segmentation.

1 Introduction

The simplest technique to segment the characters is to use inter-character gap between the characters. This technique cannot be applied on touching half characters. Also, the technique used to segment the printed characters cannot be applied to handwritten documents due to different writing styles, different sizes of characters and different shapes of characters in texts written by different people. The presence of half characters in handwritten text makes the problem of segmentation more complex.

2 Related Work

A good survey about OCR is given in [1]. Hindi is the official language of India. To the best of author's knowledge, no commercial OCR for handwritten Hindi text is available, yet. Many algorithms have been developed for segmenting touching characters in Indian scripts, but most of them are on printed text. Bansal and Sinha [2] had segmented the conjuncts (type of touching characters) based on structural properties

C. Singh et al. (Eds.): ICISIL 2011, CCIS 139, pp. 48–53, 2011.

of text in printed Devanagari script. They segmented the conjuncts with an accuracy of 84%. Jindal *et al.* [3, 4] had segmented the touching characters in middle zone and upper zone of printed Gurmukhi script using structural properties of the script. Chaudhuri *et al.* [5] had used the principal of water overflow from a reservoir to segment touching characters in Oriya script. Garain and Chaudhuri [6, 7] had used a technique based on fuzzy multifactorial analysis to segment touching characters in printed Devnagari and Bangla scripts.

Tripathi and Pal [8] had worked on segmentation of touching characters in handwritten Oriya text using structural, topological and water reservoir features. But this technique cannot be directly applied to handwritten Hindi text due to presence of half characters touching the full characters(conjuncts). The work on line segmentation, consonant segmentation, upper modifier segmentation and lower modifier segmentation in Handwritten Hindi text were explained by us in [9, 10]. In this paper, we have explained a new method based on structural features for segmentation of half characters in handwritten Hindi text.

3 Database

All experiments were conducted on database constructed by taking handwritten data from 15 writers. The handwritten documents were reduced in size in paint to 35% to increase the speed of execution. The percentage of stretching of the document in horizontal and vertical direction was same. In some documents, up to 2 degree skew correction was done in paint. This was done on whole document and not on particular lines.

Figure 1 contain part of handwritten Hindi database.

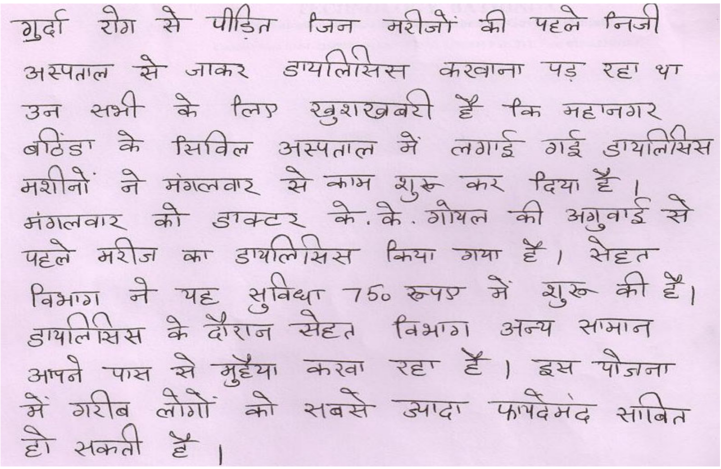
गुर्दा रोग से पीड़ित जिन मरीजों की पहले निजी
अस्पताल से जाकर डायलिसिस करवाना पड़ रहा था
उन सभी के लिए खुशखबरी है कि महानगर
बठिंडा के सिविल अस्पताल में लगाई गई डायलिसिस
मशीनों ने मंगलवार से काम शुरू कर दिया है।
मंगलवार को डाक्टर के.के. गोयल की अगुवाई से
पहले मरीज का डायलिसिस किया गया है। सेहत
विभाग ने यह सुविधा 750 रुपए में शुरू की है।
डायलिसिस के दौरान सेहत विभाग अन्य सामान
अपने पास से मुहैया करवा रहा है। इस योजना
में गरीब लोगों को सबसे ज्यादा फायदेमंद साबित
हो सकती है।

Fig. 1. Part of Database

4 Characteristics of Hindi Language

Devanagari is the script for writing Hindi language. Hindi is written from left to right and there is no concept of upper or lower case. The half characters may touch with full characters to make the characters called conjuncts.

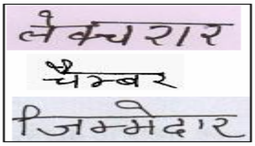

Fig. 2. Conjuncts

This paper deals with segmentation of these conjuncts. When two or more characters are combined to form a word, the horizontal lines touch each other and generate a header line called *shirorekha*. The vowels (modifiers) can be placed at the left, right (or both), top or bottom of the consonant.

5 Segmentation of Half Characters

For separation of Half characters (Conjuncts), the following algorithm has been developed.

After separating the lower and upper modifiers, the consonant separation is done. We determine vertical projection profile of the word and pass the separated characters through this algorithm. Let m is the matrix storing the conjunct character. Let r is number of rows and c is number of columns.

The determination of presence of half character is very challenging. The characters whose width is greater than 1.65 times the height of a character are assumed as conjuncts or touching characters and treated separately. We also tried the algorithm with threshold value as 1.5 but the results are best with threshold value of 1.65 only. But, if the height of a character is very small i.e height of the character is less than 12 pixels (in printed text), it will not work. To handle this problem we choose the threshold value as 1.4 for characters with less than 12 pixels in height.

The algorithm is as follows:

Step 1: For each column (i), the number of pixels are determined from row (r/7) to rth row, and stored in an array say vpixels(i), i=1 to n, where n is the number of columns.

Step 2: Starting from the left most pixel, we scan the character from left towards right upto first 70% part of the character i.e $(c\times0.7)^{th}$ column of the character. If number of pixels vertically in two continuous columns is greater than one and column position is less than ceil(c/5), we set the flag flag_1 and continue to scan further till we get two continuous columns with single pixel. For printed text(r<12), the flag is set if number of pixels in any column is greater than one and column position is less than ceil(c/5).

Step 3: If we get two continuous columns with single pixel and column position is greater than $(c/5)^{th}$ column we set another flag flag_2 and continue to scan towards right till $(c\times0.7)^{th}$ column.
Step 4: If flag_1 and flag_2 are set and we get the column with more than one pixel again, and column is between $(c/4)^{th}$ and $(c\times0.7)^{th}$ column, we store the column-1 position in a variable say v1.
Step 5: In this step following conditions are checked to avoid over segmentation of other characters that satisfy the condition that their width is greater than 1.65 times the height of a character like , , , etc., which are generally written longer in width while writing the text.

i) The number of pixels in v1 column is less than half of the maximum no. of pixels in any column or there are more than three continuous columns with one pixel
ii) The maximum height of remaining columns(v1+1 to c) is greater than or equal to maximum height of any of the columns from 1 to v1.
iii) Presence of two continuous columns with pixels greater than two from v1 to v1+4 columns.

If all the above three conditions are true, we store the value of v1 for half character separation otherwise it is set to zero. If v1 is zero, than it is a problem of second type i.e. two consonants are touching each other (Segmentation is done separately).
Step 6: Starting from first left most pixel to the column v1, we copied the matrix to another matrix say m1 and copied the pixels from column v1 to end column, to matrix m2.

The step 2 is further modified to solve the problem of pen width in the starting of character. Instead of scanning the character from left most pixels we scan from third pixel position.

6 Results

The proposed algorithm is tested on both handwritten as well as on printed Hindi text to segment the half characters and it gives very good results (Table 1 and Table 2, figure 3).

Table 1. Accuracy of Segmentation of Handwritten Hindi Text

Total Words	Total Half Characters	% of Half Characters	Half Characters Correctly Segmented	% Accuracy of Half Character Segmentation
1294	106	8.18	88	83.02

Table 2. Accuracy of Segmentation of printed Hindi text

Total Words	Total Half Characters	% of Half Characters	Half Characters Correctly Segmented	% Accuracy of Half Character Segmentation
345	24	2.9	21	87.5

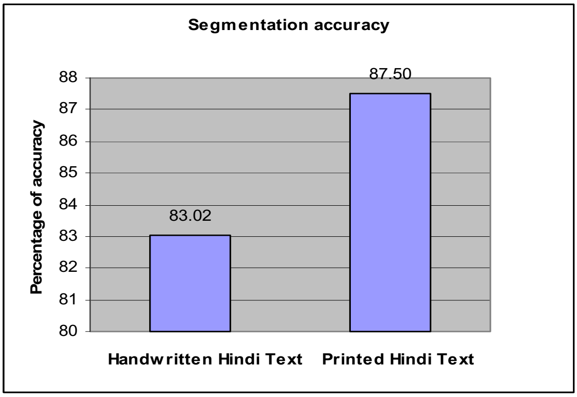

Fig. 3. Half Character Segmentation Results

Further, for visual inspection, some of the correctly segmented conjuncts are shown in figure 4 and incorrectly segmented conjuncts in figure 5.

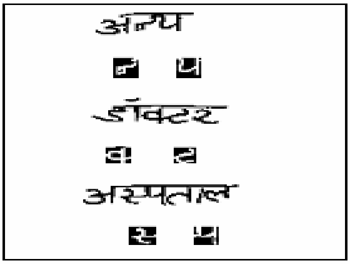

Fig. 4. Correctly segmented conjuncts

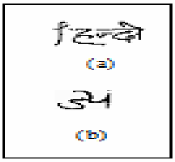

Fig. 5. Incorrectly segmented figures

In figure 5(a), the half character is overlapped with consonant and not attached with during vertical separation of consonants. In figure 5(b), the half character and character are very much overlapped and no vertical single pixel columns present as required in step 3.

The main problem in half character separation is the overlapping of half character with the full character. The determination of presence of half character is even more difficult task to handle.

Multiple conditions specified in step five are put on data to avoid the over segmentation of other characters (like) with width greater than 1.65 times the height of a character. The same technique is also tried on printed Hindi text.

7 Discussion

From the above results, it is clear that the proposed technique used to segment the conjuncts (half characters) in Handwritten Hindi text is very useful. The study may be carried out in future in the following direction:

1. The above algorithm with some modification may be used to segment the touching characters in handwritten Hindi text.
2. The above technique may be used to segment touching characters in other Indian scripts.

References

[1] Mori, S., Suen, C.Y., Yamamoto, K.: Historical review of OCR Research and development. Proceedings of the IEEE 80(7), 1029–1058 (1992)

[2] Bansal, V.: Integrating knowledge sources in Devanagari text recognition. Ph.D. thesis, IIT Kanpur, INDIA (1999)

[3] Jindal, M.K., Lehal, G.S., Sharma, R.K.: On Segmentation of touching characters and overlapping lines in degraded printed Gurmukhi script. International Journal of Image and Graphics (IJIG) 9(3), 321–353 (2009)

[4] Jindal, M.K., Sharma, R.K., Lehal, G.S.: Segmentation of Touching Characters in Upper Zone in printed Gurmukhi Script. In: Proceedings of the 2nd Bangalore Annual Compute Conference, Bangalore, vol. (9). ACM, New York (2009)

[5] Chaudhuri, B.B., Pal, U., Mitra, M.: Automatic recognition of printed Oriya Script. In: Int. Conf. on Document Analysis and Recognition, pp. 795–799 (2001)

[6] Garain, U., Chaudhuri, B.B.: Segmentation of touching characters in printed Devnagari and Bangla scripts using fuzzy Multifactorial Analysis. IEEE Trans. on Systems, Man and Cybernetics. Part C 4(32), 449–459 (2002)

[7] Garain, U., Chaudhuri, B.B.: On recognition of touching characters in printed Bangla documents. In: Int. Conf. on Document Analysis and Recognition, Germany, pp. 1011–1016 (1997)

[8] Tripathi, N., Pal, U.: Handwriting segmentation of unconstrained Oriya Text. Sadhana 6(31), 755–769 (2006)

[9] Garg, N.K., Kaur, L., Jindal, M.K.: Segmentation of Handwritten Hindi Text. International Journal of Computer Applications (IJCA) 1(4), 22–26 (2010)

[10] Garg, N.K., Kaur, L., Jindal, M.K.: A new method for line segmentation of Handwritten Hindi Text. In: Proceedings of the 7th International IEEE Conference on Information Technology: New Generations (ITNG), pp. 392–397 (2010)