# FRAGMENTATION OF HANDWRITTEN TOUCHING CHARACTERS IN DEVANAGARI SCRIPT

Shuchi Kapoor[1] and Vivek Verma[2]

[1,2]Department of Computer Engineering, Rajasthan College of Engineering for Women, Jaipur, Rajasthan

***ABSTRACT***

*Character Segmentation of handwritten words is a difficult task because of different writing styles and complex structural features. Segmentation of handwritten text in Devanagari script is an uphill task. The occurrence of header line, overlapped characters in middle zone & half characters make the segmentation process more difficultt. Sometimes, interline space and noise makes line fragmentation a difficult task. Sometimes, interline space and noise makes line fragmentation a difficult task. Without separating the touching characters, it will be difficult to identify the characters, hence fragmentation is necessary of the touching characters in a word. So, we devised a technique, according to that first step will be preprocessing of a word, than identify the joint points, form the bounding boxes around all vertical & horizontal lines and finally fragment the touching characters on the basis of their height and width.*

***KEYWORDS***

*Fragmentation, Joint point algorithm, Shirorekha, side bar, middle bar .*

## 1. INTRODUCTION

After many years of research, hand written touching character segmentation remains a challenging task. Touching character patterns emerge when two adjacent characters are written too close, therefore, some parts of character are connected horizontally/left-right. In printed script generally two consecutive characters do not overlap or touch but in handwritten everyone is having their own writing style. In this present work we have proposed a method of segmenting characters from handwritten Devanagari words. There is unpredictable variation in handwriting style from person to person and Devanagari having large number of characters set makes segmentation of handwritten words complex. Papers [1,2] give some surveys on segmentation techniques for machine printed text. For segmentation of handwritten text also a survey paper [3] is available. Very few papers are available on segmentation of handwritten characters from word. No work for touching handwritten Devanagari character segmentation is found so far to the best of our knowledge. In [4], Utpal Garain and Bidyut B. Chaudhuri, presents a new technique for identification and segmentation of machine printed touching characters. They devised a fuzzy multifactorial analysis technique which has been applied to printed documents in Devanagari and Bangla. In [5], Veena Bansal and R.M.K.Sinha depict the system architecture and its components for Devanagari text recognition system. The system consists of a solution blackboard and various knowledge sources. In [6], M. Hanmandlu and Pooja Agrawal made an attempt to segment the handwritten Hindi words. The segmentation problem is computed due to different writing styles and presence of modifiers on all sides of characters. So, they proposed a structural approach to identify the similarities and differences between structure classes. Veena Bansal and R.M. K. Sinha [7], presents an algorithm for segmentation of machine printed touching Devanagari

DOI : 10.5121/ijitmc.2014.2102 

characters (also referred to as conjuncts) into its constituent symbols and characters. In the Existing system the selective algorithm is developed for the identification and removal of header line and for further segmentation from the handwritten Devanagari characters. Proposed algorithm extensively uses structural properties of the script. It uses vertical projection, continuity of collapsed horizontal projection. We use structural properties of Devanagari script for fragmentation of characters.

## 1.1. Characteristics of Devanagari Script

Devanagari Script is the most common script in India which include Marathi, Hindi, Sanskrit & Nepali. It has 12 vowels and 35 consonants as shown in figure.1. Vowels can be written isolated or by using variety of modifiers overhead, down, foreword or backward after the consonants they belong to & those characters are called conjuncts as shown in figure 2. At times new shapes are formed after combining two or more consonants & these new shapes are called compound characters as shown in figure 3. In Devanagari script there is a header line known as Shirorekha, this feature distinguish Devanagari script from English, so English will be extracted from these script. There are four imaginary lines that drawn for a word, Headline is s same as header line, Baseline where characters completed but excluding below modifiers. After below modifiers line drawn called Lowerline and the line above Headline after modifiers are called Upperline resultant text is partitioned into 3 zones- Upper zone between Upperline and Headline, Middle zone between Headline and Baseline & last is Lower zone between Baseline and Lowerline.

A typical zoning is shown in Figure.4.

In Hindi we have three types of characters as follows also shown in Figure 5a,5b, 5c

1.END-BAR Characters
2.MIDDLE-BAR Characters
3.NO-BAR Characters

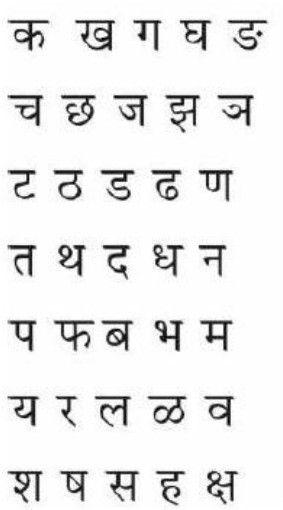



Figure 1. Consonants.

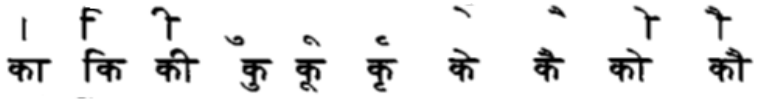


Figure 2. Modifiers(Ascenders and descenders)

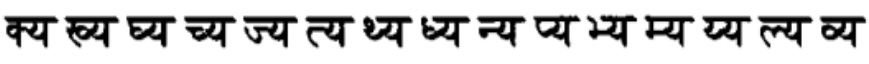


Figure 3. Compound Characters

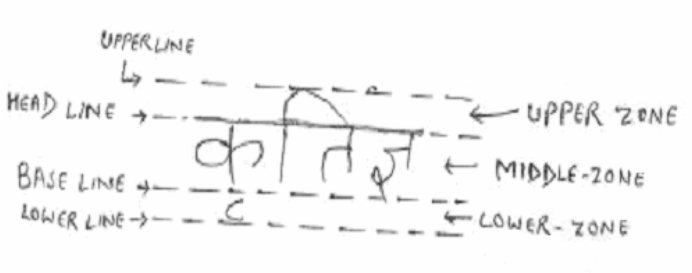



Figure 4. Different Zones of Devanagari Text.

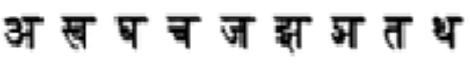


Figure 5a. END-BAR Characters

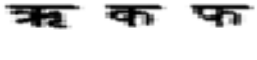


Figure 5b. MIDDLE-BAR Characters

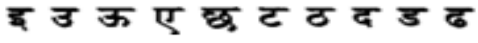


Figure 5c. NO-BAR Characters

As we are considering handwritten Devanagari, it again adds up complexity because while writing, characters of the word may touch each other at different position as shown in figure 6. For recognition, it is necessary to segment these touching characters of words. For this we propose a system.

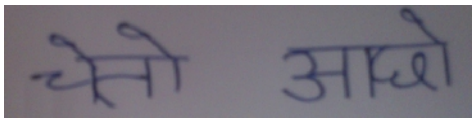
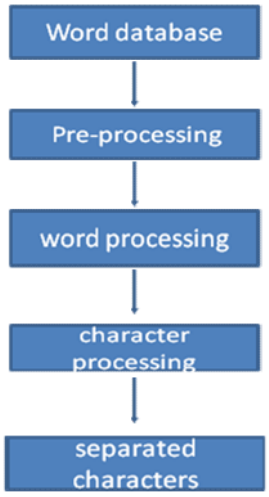


Figure 6. Samples of Touched Characters

# 2.RESEARCH METHOD

In Devanagari script few categories are classified on the basis of structural properties that are given in figure 5.

**Proposed System**




## 2.1. Pre-processing:

1. In Image Acquisition scan a document and storing it as an image. This will act as input to the program. Higher resolution produces better results.
2. In every binary image, each pixel is in form of two value: 1 or 0. The input image is first converted into binary or image will be complemented means background is black and foreground i.e words are white and represented as 1's. In binary image attached words are found and isolate each word separately.
3. In binarized image all small black patches are removed because that spots can create confusion while thinning process. Bounding boxes are formed around small black spots and converted into white.

**Protruding points**

1. Protruding points are those unnecessary joint points that generate after thinning process. So, these protruding points are removed so that actual joint points can be generated between those actual line exists.
2. In 3x3 matrix sum will be calculated, around each pixel, minus centre pixel. If this sum is less than or equal to 3 then the protruding pixel is removed. This process is visualized in the figure.
3. Image with protruding points highlighted with red.

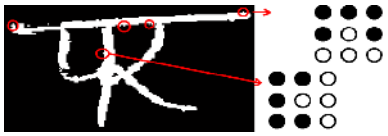


Figure 7. For eliminating unusable joint points in the image protruding points are removed

## 2.2. Word Processing Stage

### 2.2.1 The joint points

(i) Where two lines meet that point is called joint points. The lines can be straight or curved. On the basis of these points characters are dissected but here we are using joint points just to form the vertical bars and header lines.
(ii) By analyzing the design of pixels in 3x3 matrix adjacent to each pixel joint points are found.
(iii) With the help of joint points we will find vertical bars and header lines.
(iv) In 3 x 3 matrix sum will be calculated around each pixel minus centre pixel and the pixel is considered as a joint point if sum is equal to more than 3. The result 3 means three lines meeting at that point.
(v) If pixel is having only one line joining the point i.e only 1 pixel in the surrounding is called terminating points.

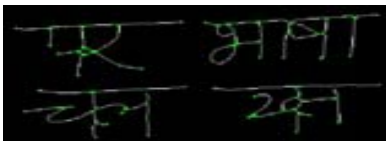


Figure 8. Joint points (shown in green)

### 2.2.2 Form Bounding Box

To fragment each character in a word bounding boxes are formed in order to identify the header line and vertical bars. Within each word horizontal rectangles are identified. In thinned image bounding boxes are formed after removing the joint points.

### 2.2.3 Removal of Shirorekha

(i) The first aim is to find and remove shirorekha.
(ii) In thinned image bounding boxes are obtained leaving joint points.
(iii) Horizontal bounding boxes are recognized in every word.
(iv) Horizontal rectangles are form on the basis of width-height ratio i.e should be more than or equal to 3:1.
(v) Heights of all pixels inside these rectangles are averaged.
(vi) If maximum number of pixels on one side of this averaged line than that rectangle are considered as shirorekha.
(v) These rectangles are removed to remove the shirorekha**.**

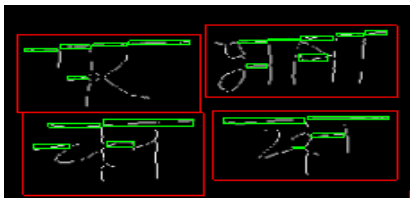


Figure 9. Horizontal Bounding Boxes

### 2.2.4 The vertical bar detection

(i) Vertical rectangles will help to detect the vertical bars.
(ii) Vertical rectangles are having width-height ratio less than or equal to 1:3.
(iii) To identify the vertical bars percentage is calculated on the basis of how much percentage of white pixels (parts of vertical bars) is coming in vertical rectangles and if it is more than 60% then the rectangle holds vertical bar.
(iv) Rectangles are stretched upto the shirorekha in the top and at the bottom & will get a new image.
(v) New image contains bounding boxes.

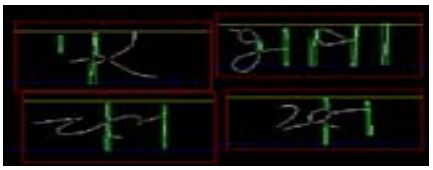


Figure 10. Vertical bars are detected

## 2.3. Character Processing Stage

*Shirorekha* and vertical bar removed image is considered for character processing.

1. Category 1: Mid Bar

Form a bounding box and check width of bounding box. For category 1, if width of bounding box is greater than single character width (means two characters are touching) then make a cut at leftmost joint point of large bounding box. We get separated characters. We get separated characters.

Figure 11. Example of category 1 for cut line finding.

2. Category 2: Side Bar

When two side bar characters are touching each other, form bounding box. If there is no large bounding box and no equal size bounding boxes, then make a cut line after each detected vertical bars.

Figure 12. Fragmentation of two side bar touching characters

3. Category 3: Non Bar

If character without Vertical Box then make cut at the right of Bounding Box.

Figure 13. Fragmentation of two non bar touching characters.

## 3. RESULTS AND ANALYSIS

Different document consists of words are given as input to proposed system. Average result for touching character segmentation of Devanagari words is 71%. Accurate character segmentation having side bar is 76%, middle bar 81%, no bar 75%.

### 3.1. Result of Touching Character Fragmentation

We consider four documents where various characters from words are touched each other in different way. Some can overlapped with next character, touched at the different position. Result of touching character segmentation shown in table 1 consists of words are given as input to proposed system. As we segments characters from words based on different categories (middle bar, side bar, non-bar). So Table 2 shows Segmentation result of different character category.

Table 1. Fragmentation result of touching characters

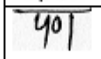
| | | |
|---|---|---|
| Middle bar character |  |  |
| |  |  |
| Side bar character |  |  |
| |  |  |
| Non bar character |  |  |
| |  |  |

Table 2. Segmentation Result of Different Category Character

| Images | Characters Category | Total | No of characters segmented correctly | % Result |
|---|---|---|---|---|
| a. | Sidebar | 25 | 16 | 64% |
| | Middle bar | 03 | 03 | 100% |
| | No bar | 08 | 07 | 87% |
| b | Sidebar | 25 | 17 | 68% |
| | Middle bar | 01 | 01 | 100% |
| | No bar | 13 | 09 | 69% |
| c | Sidebar | 10 | 10 | 100% |
| | Middle bar | 02 | 01 | 50% |
| | No bar | 02 | 02 | 100% |
| d | Sidebar | 19 | 15 | 78% |
| | Middle bar | - | - | - |
| | No bar | 05 | 03 | 60% |
| e | Sidebar | 13 | 08 | 61% |
| | Middle bar | 04 | 01 | 25% |
| | No bar | 04 | 03 | 75% |
| f | Sidebar | 13 | 11 | 84% |
| | Middle bar | 05 | 03 | 50% |
| | No bar | 05 | 04 | 80% |

### 3.2. Result of Handwritten and Printed Document

We tested our system on handwritten and printed document without touching characters.

**Handwritten Document**

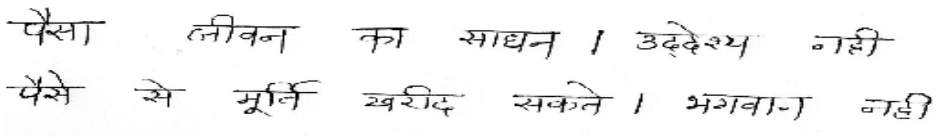


Figure 14. Handwritten Document without touching characters

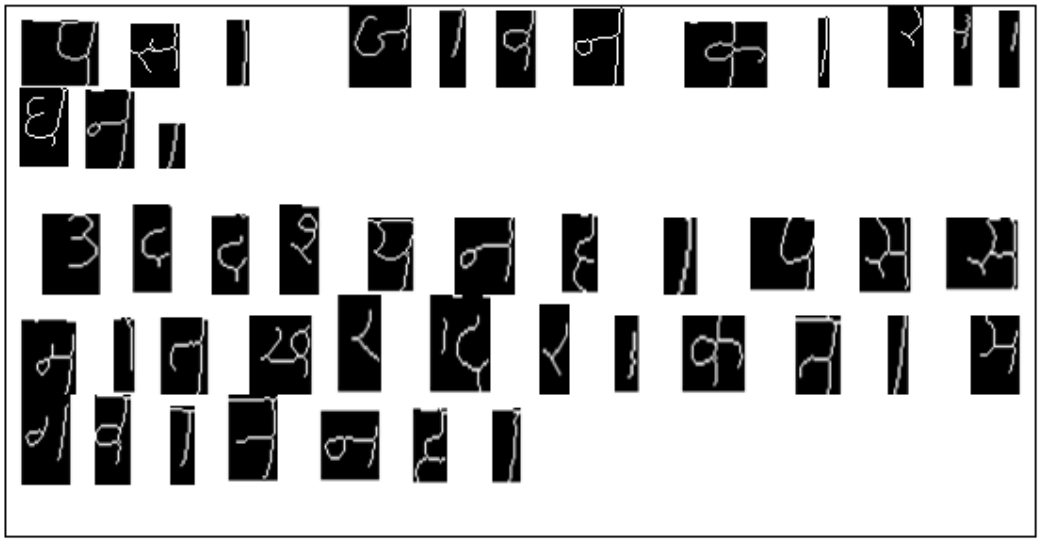


Figure 15. Character Fragmentation of Handwritten document

## Printed Document

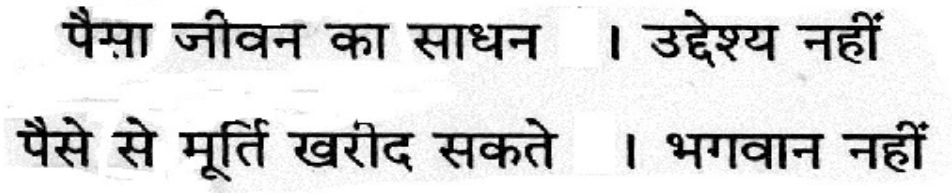



Figure 16. Printed Document without touching character

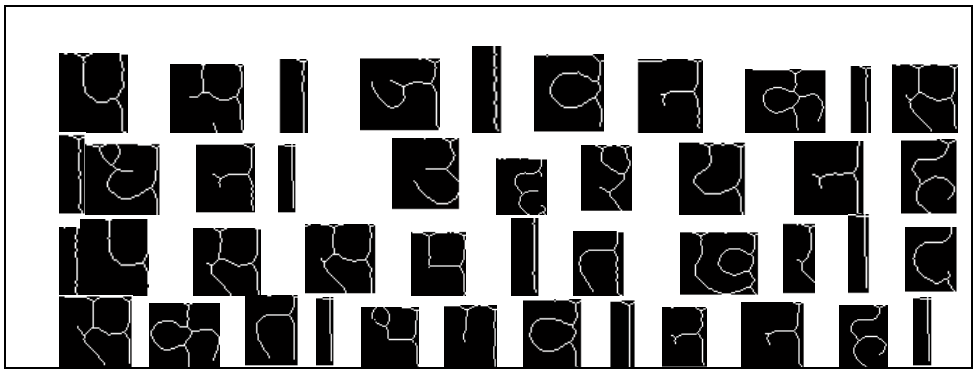


Figure 17. character Fragmentation of printed document

Table 3. Character Fragmentation Result of Handwritten and Printed Documents

| Imags Without Touching Characters | No of Characters in Middle Zone including Matra | No of Character Correctly Fragmented | % Result of Correctly Fragmented Character |
|---|---|---|---|
| Handwritten document | 43 | 40 | 93% |
| Printed document | 43 | 43 | 100% |

### 3.3.Failure Reason for Characters Segmentation

There are some characters and writing style which degrade the performance of our system. Table 4 gives reasons for failure of some characters

Table 4. Reason for Failure of Some Characters Segmentation

| Characters | Segmentation Result | Reason |
|---|---|---|
| ग ण |  | • Space between parts of characters.<br>• Our system considered first part of character as non-bar character make cut after that. |
| क |  | • Writing style of क is different around the curves of character.<br>• Our system put a constraint that area difference of two curves should be less than word height (blevel-slevel).<br>• Curve shape should be start from Shiro-rekha and less than 25% of word height. |
| भ म |  | • First part of these character consist of pixel more than 60% then considering as vertical bar and make cut after that. First part should be less than side bar |
|  |  | • Overlapping of characters.<br>• Cut some portion of characters. |

## 4. CONCLUSION

We are successfully able to find joint points between characters of a word which makes us easy to find vertical lines and horizontal lines of characters. We tried to do segmentation without skew correction. Average result of handwritten touching characters from words is 71%. Accurate character

segmentation having side bar is 76%, middle bar 81%, no bar 75%. For non-touching printed documents and handwritten document it gives us 100% and 93%result respectively.This method fails for certain characters because of constraints that we decide will not match always to a lot of variation in handwriting i.e. , height of two vertical bar characters i.e.Characters like causes problem as gap between parts of characters that can be solved by the recognition system because our system consider first part as non-bar character and cut after that. As the future work i s conc e rn, we pass these separated characters as input to the character recognition system. Again we have detected baseline, header line of word which will again useful for the recognition system to detect lower modifier and upper modifier. The character separation technique explained above can be applied on other Indian scripts also.

## ACKNOWLEDGEMENTS

I sincerely feel that the credit of project work could not be narrowed down to only one individual.This work is an integrated effort of all those concern with it, through whose able cooperation and effective guidance I could achieve its completion.I express my gratitude to my guide Assistant Prof. Vivek Verma, our co-ordinator Assistant Prof. Vandna Verma and HOD Assistant Prof. Rakesh Sharma for being kind enough to spare his valuable time, provide insights and guidance on a complex topic.I am also thankful to Principal and management of RCEW for their support and encouragement.I also thank computer laboratory staff for their valuable support

## REFERENCES

[1] Y. Lu, "Machine Printed Character Segmentation – an Overview",Pattern Recognition, vol. 28, No. 1, pp. 67-80, 1995.
[2] R. G. Casey, E. Lecolinet, "A Survey of Methods and Strategies in Character Segmentation", IEEE, vol. 18 No. 7, pp. 690-706, July 1996.
[3] C. E. Dunn, P. S. P. Wang, "Character Segmentation Techniques for Handwritten Text-A Survey", Proc. 11th Int.Conf.on Recognition Methodology and Systems, vol. II, pp.577-580, 30 Aug.-3 Sept.1992.
[4] Utpal Garain and Bidyut B. Chaudhuri, "Segmentation of Touching Characters in Printed Devnagari and Bangla Scripts Using Fuzzy Multifactorial Analysis", IEEE transactions on systems, man, and cybernetics—part c: applications and reviews, vol.32, 4, November 2002, 449-459.
[5] Veena Bansal and R. M. K. Sinha,"Integrating Knowledge Sources in Devanagari Text recognition System", IEEE transactions on systems, man, and cybernetics—part a: systems and humans, vol.30, no. 4, July 2000, 500-505.
[6] M. Hanmandlu, Pooja Agrawal, "A Structural Approach for Segmentation of Handwritten Hindi Text", Proceeding of the International Conference on Cognition and Recognition, Mandya, Karnataka, India, 22-23 December 2005, 589-597.
[7] Veena Bansal and R. M. K. Sinha,"Segmentation of Touching and fused Devanagari Characters", Pattern recognition, vol. 35:2002, 875-893.

### Authors

She received her B.E degree from J.E.C.R.C, Jaipur(Rajasthan University) in 2005.She is pursuing her M.Tech in Computer Science from R.C.E.W(Rajasthan Technical University). She has presented and published papers in 4 conferences. Her areas of interest are image processing and networking.

He is an Assistant Professor at Rajasthan College of Engg For Women. He received his B.Tech degree from Rajasthan Technical University. He did his M.Tech in Computer Science from C-DAC Noida (GGSIPU Delhi). He has several International and National publication in SCI journals.