A Technique for Segmentation of Handwritten Hindi Text

Ms. Vaishali G. Bhujade[1], Ms. Chhaya M. Meshram[2]
[1]Department of Computer Engineering, [2]Department of Information Technology
[1]B.D.C.O.E. Sevagram, Wardha (M.S.), [2]B.D.C.O.E. Sevagram, Wardha (M.S.)

***Abstract*— The main purpose of this paper is to provide the new method for Segmentation of Handwritten Hindi text. Segmentation is one of the major stages of character recognition. The method is based on header line detection, upper modifier detection, lower modifier detection, character detection and contour following technique. Paper introduced the new concept of resizing the image, so that system able to reduce space as well as time complexity. Characters can be in arbitrary location, scale and orientation. There are three steps in recognition which are pre-processing, feature extraction and classification. Firstly, the contour following after header line detection correctly separates upper modifier then character and lower modifier. Secondly, this paper provides a brief review of text line segmentation techniques for handwritten text which can be very useful for the beginners who want to work on conversion of Hindi characters to English characters. This concept of character categorization and recognition using Java as a base can be a very major step in the field of handwritten text translator.**

Keywords— character segmentation, header line detection.

I. INTRODUCTION

Segmentation of a document into lines and words and of words into individual characters and symbols constitute an important task in the optical reading of texts. Presently, most recognition errors are due to character segmentation errors. Very often, adjacent characters are touching, and may exist in an overlapped field. Therefore, it is a complex task to segment a given word correctly into its character components. In this paper, we have considered the problem of segmenting printed text in Devanagari. Devanagari is the script for Hindi (the official language of India), Sanskrit, and Marathi and Nepali languages, among others. It is used by more than 400 million people across the globe. Devanagari is a derivative of ancient Brahmi, the mother of all Indian scripts. It is a logical script composition of symbols in two dimensions. The two dimensional composition has to be decomposed into symbols that are meaningful in the script.

Off-line handwritten Hindi character recognition is one of the most difficult tasks of optical character recognition because of complex patterns, large number of classes involving basic characters and matras, different writing styles and sizes etc. It also requires a large amount of time as recognition module has three stages viz., pre-processing, feature extraction and matching with the database. We adopt the divide-and conquer policy wherein a major category is divided into sub-categories thus making the classification process simpler. Accordingly we are inclined to develop a technique to divide a complete set of characters into some sub sets using a similarity measures. It may be noted that the classification of handwritten Hindi characters into sub-groups has been a challenging problem since the handwritten characters do not have a fixed size and shape. So they are quite different from the printed characters. In the case of printed characters, vertical bar occupies a single column whereas handwritten characters might occupy more than one column. Moreover the header line is never straight.

Handwritten character recognition is difficult task compared to machine printed character recognition in the area of Optical Character Recognition. Devanagari is a script for Hindi text. A lot of research is done on the printed Hindi text, but less work has been done on the handwritten Hindi text recognition. It has three major steps such as pre-processing, segmentation and recognition. Segmentation is the important step. Segmentation also contains three major steps such as line segmentation, word segmentation and character segmentation. If we fail in doing line segmentation then entire segmentation process goes wrong. A lot of research has been done in the past on line segmentation of handwritten text. A wide variety of line segmentation methods for handwritten
Documents are reported in the literature. Some of the important methods for line segmentation are projection based method [1] and [2], Hough transform based method [3], smearing method [4], grouping method [5], graph based method [6], CTM (Cut text Minimum) approach [7], Block covering method [8], linear programming method [9] and curve based [15].

The projection based methods are successful in the case of straight and easily separable lines only. A lot of research is going on how to detect overlapped lines and characters. The method which is based on header line and base lines detection and average line height is assumed as 30 pixels gives good results in case of fixed resolution images. The main purpose of this paper is estimating the average line height and based on it, finding the header lines and base lines Word segmentation is very easy because gap between words is generally more, so we can separate them easily. But in

Character segmentation most of the researchers use vertical projection method [11]. Some researchers use Hidden Markov model [13][14]. But these methods do not work well for the overlapping characters. In next Section, we have discussed the characteristics of Hindi language. In Section 2, segmentation techniques used for segmenting the handwritten Hindi text have been discussed. Finally, Section 3-4 contains experimental results and discussions.

II. SEGMENTATION

Segmentation is one of the most important phases in character recognition process. Segmentation is the process of segmenting the whole document image into recognizable units. The segmentation process is divided into four major parts.

i. Header Line Identification and Removal
ii. Segmentation of Upper Modifiers (Top Strip) and Character-Lower Modifiers (Core-Bottom Strip)
iii. Identification Whether the Segmented Character Contains a Lower Modifier
iv Detecting and Subsequently Segmenting Lower Modifiers from Characters Containing a Bar
v. Segmenting Lower Modifiers from Characters without a Bar

A. Header Line Identification and Removal

The header line is the most visible and distinguishing part of a word. By separating the header line we can obtain the top and core-bottom parts of a word. For separation of header line, the horizontal density (i.e., the number of pixels in each row) of the word is calculated and the region with maximum density lying within the top 3/4th of the word is identified. We have considered the top 3/4th of a word because of two reasons: (i) Some words, shown in Fig 1(b), have a high horizontal density in some parts (lower 1/4th of word), hence by considering the top 3/4th of the word such areas are isolated, (ii) At times, the upper modifiers lie within the top half of the word. Taking the top 3/4th area will allow us account for such samples.

Since the header line covers the entire word, the region with the highest pixel density will give us the position of the line. In the handwritten characters the header line covers multiple rows in contrast to printed characters whereas it covers a single row. Once the position of header line is determined, we remove it by changing the gray level of 3 rows (header line, and one row above and one row below it) into the background gray level as shown in following Fig

Fig. 1(a) Original word (b) Without header line word image

B. Segmentation of Upper Modifiers (Top Strip) and Character-Lower Modifiers (Core-Bottom Strip)

For segmentation of the top strip and core-bottom strip, a contour-tracing algorithm is applied. The algorithm needs the coordinates of an image pixel that lies on the contour and returns the positions (row, column) of all the connected points by checking the continuity of the input pixel around its 3x3 neighbourhood. In the top strip, the position of the first black pixel (from the top left corner) is passed on to the contour tracing function. The function returns the positions of all the pixels in the first upper modifier.

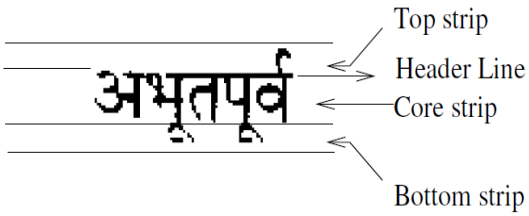

Fig 2 Devnagri text sample

Using these positions the modifier is separated from the top strip and the corresponding area is whitened. Similar process is carried out on the rest of the top strip to separate out all the upper modifiers. To separate characters (simple/conjunct with lower modifiers) from the core-bottom strip, similar process is repeated. We then take the original word image with the header line present but without the upper modifiers. The character images are separated out from this word image yielding the segmented character image with the header line. Each separated character (with header line) or modifier is saved as a new image.

C. Identification Whether the Segmented Character Contains a Lower Modifier

Before identification of lower modifiers, Devanagri characters can be divided into three groups based on the presence of the vertical bar, namely, the end bar characters: म, ज, त, ल, न, च, स, ब, श्र, ज्ञ, य, प middle bar characters: क, फ and non-bar characters: ह. र. इ. द. ट. ठ. छ. ड.. To determine the presence and position of a vertical bar the segmented character image is divided into 3x3 windows, as shown in Fig 3(a). Figure 3(b, c, d, e) shows the characters that have a vertical bar in the end, or in the middle, or do not have a vertical bar. To detect a middle vertical bar, we examine boxes 2 and 5 and check whether they contain more than 90% of the rows that are black. If so, it confirms the presence of a bar. For detecting the presence of an end bar, we similarly examine the boxes 3, 6 and 9.

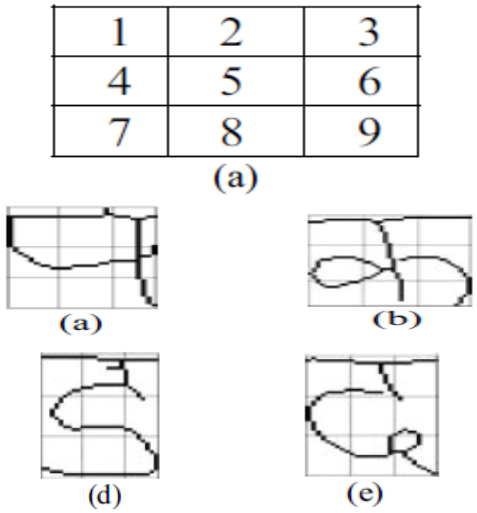

Fig 3: (a) 3×3 window & End-bar (pa) (b) Middle-bar (ka) (d)&(e) Without-Bar (da) characters with 3×3 window

D. *Detecting and Subsequently Segmenting Lower Modifiers from Characters Containing a Bar*

To detect if a character has a lower modifier shown in Fig. 4(a), system examine the lower half of the image. We start scanning row-wise from the top and after detecting a single black pixel per row, we look for more than one pixel in a row. If so, this indicates the presence of a lower modifier, else the character is either a simple character or a conjunct character. The row at which we first encounter multiple pixels is taken as a part of the lower modifier and the position of the black pixel is then recorded. In the row above this, the pixels are cleared (made white) as shown in Fig. 4(b). The recorded position is passed on to the contour-tracing program that returns the positions of the pixels belonging to the lower modifier. The character devoid of the modifier is shown in Fig. 4(c) and the modifier can then be separated from the character as in Fig. 4(d).

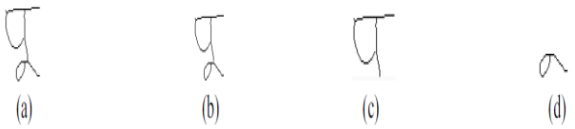

Fig. 4 (a) Handwritten End bar-character (i) with lower modifier (w), (b) After detection of lower modifier, image shows single cleared pixel, (c) Segmented character (i), (d) Segmented lower modifiers (w).

E. *Segmenting Lower Modifiers from Characters without a Bar*

To detect the lower modifiers, we start scanning row-wise from (rows/2) to (rows-3) in the word image and check for a case where we get more than one pixel in a row after having encountered a single black pixel per row. This situation indicates the presence of the lower modifiers as shown in Fig. 5(a). Otherwise the character is either simple or conjunct. But in some of the characters like B, N, b, n we may encounter problems, so to solve these problems in the characters like n, we take recourse to the different concept. If the sum of pixels in the last row of such

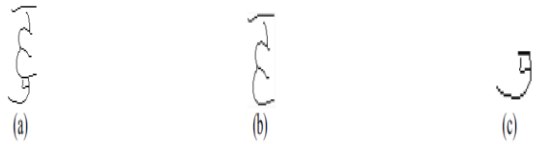

Fig. 5 (a) Handwritten character without bar (g) with lower modifier (q), (b) Segmented character (g), (c) Segmented lower modifier (q).

character image is less than 3 and if the box 9 of Fig. 3(a) has more than 90% of the row pixels, this indicates that the character " n " does not have any lower modifier otherwise it has lower modifier as shown in Fig.3.6. But characters such as (B, N) face some problems. For separation of the lower modifiers from the characters, we consider the top half part of the lower half image having lower modifier and start scanning from top to bottom to find the most horizontal dominating line in this window. Then we check one row below this most dominating row and record. If pixels in this row (which is below that most dominating row) are connected then the pixels of that most dominating row are cleared (made white). The recorded positions are passed on to the contour tracing program and the program returns the position of the pixels belonging to the lower modifier as shown in Figures 5(c) and 6(c). Remaining portion (after separation of lower modifiers) of the character image having lower modifier is either a simple or conjunct character. If the row below the most dominating row is not connected, then the pixels of the row, which is above the most dominating row, is taken and cleared (made white). The position of the pixels of the most dominating row is recorded. The recorded positions are passed on to the contour tracing program and the program returns the positions of the pixels belonging to the lower modifier. Figures 5(b) and 6(b) show the segmented character.

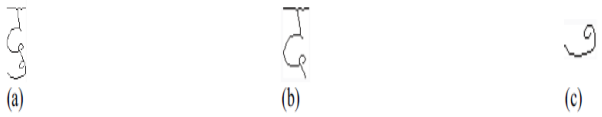

Fig. 6 (a) Handwritten character without bar (n) with lower modifier (q), (b) Segmented character (n), (c) Segmented lower modifier (q).

III. METHODOLOGY FOR SEGMENTATION

Following are the different algorithms for above mentioned major parts of segmentation.

A. *Header Line Identification and Removal*

For header line identification and removal, the horizontal projection method is used. The algorithm has following steps:

Step 1: Read the given image. Store the no. of black pixels found in each row.

Step 2: Identify the row which contains maximum black pixels.

Step 3: Store that row no. into any variable.

Step 4: Convert the black pixels of previous row, that row and next row into white pixels.

Step 5: Create the new image using the above data with newly added white pixels which replaces black pixels. Created image is nothing but Image without Header Line.

B. Segmentation of Upper Modifiers

Algorithm for segmentation of upper modifiers after header line detection has following steps:

Step 1: Read the position of header line which is already determine in Header Line Identification and Removal algorithm.

Step 2: Cut the original image from top up to the position of Header line.

Step 3: Create the new image using above data, which contain upper modifiers.

C. Character Segmentation

For character separation the vertical projection method is used after header line detection. The algorithm has following steps:

Step 1: Image without upper modifier and without header line is inputted to this method.

Step 2: Scan the image vertically. Find the columns with very less black pixel density.

Step 3: Separate the each and every character using that column numbers which identified in previous step. Store each in to characters array.

D. Identification and Separation of Lower modifiers:

The algorithm for identification & separation of lower modifiers has following steps:

Step 1: To separate lower modifiers, first find the difference in heights of Characters. If difference between maximum height and minimum height is at least 15% of the height, then assume lower modifier exists otherwise not.

Step 2: Then separate the lower modifier from the row of the image from which we get the difference in the content of the current row pixels and next row pixels is found more than threshold value. Store the lower modifier in array which stores lower modifiers. At the same time store that character without lower modifier in separate array.

IV. EXPERIMENTAL RESULTS

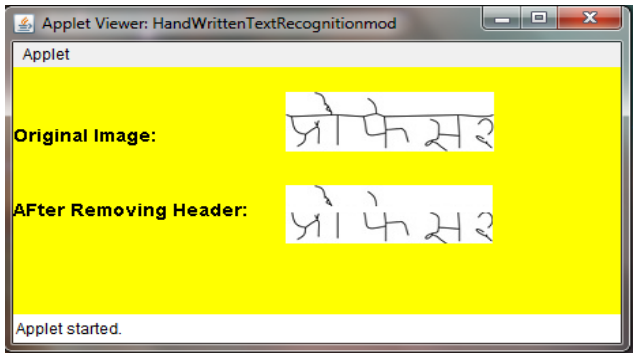

Fig. 7 Header line Identification and Removal

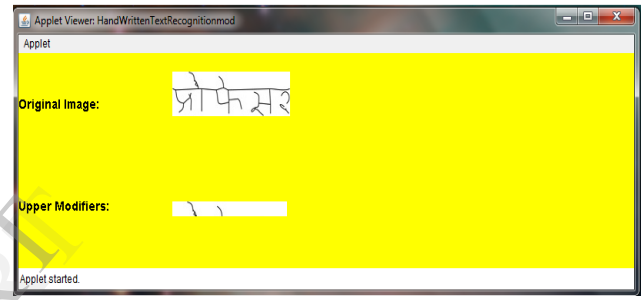

Fig. 8 Segmentation of Upper Modifier

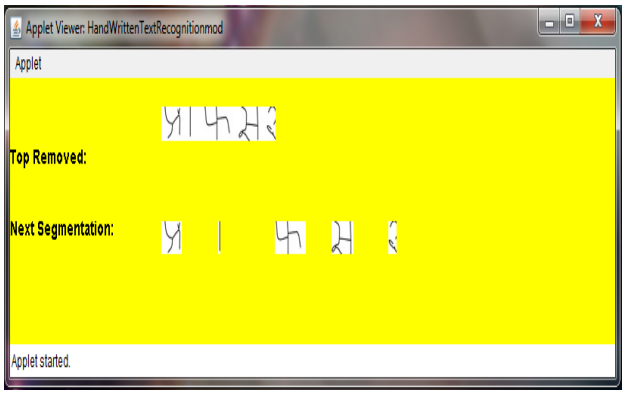

Fig.9 Segmentation of Characters

Fig.9 Identification and Separation of
End Bar, Middle Bar and No bar Characters

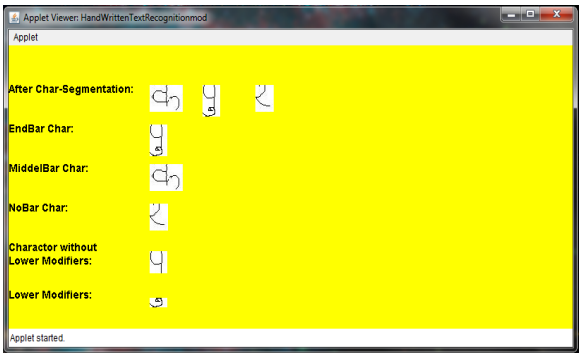

Fig.10 Identification and Separation of End Bar, Middle Bar, No bar Char, Char without Lower Modifier and with lower Modifier

V. CONCLUSION

Segmentation of handwritten Hindi text is a complex task as matras and the header line are responsible for the complexity. This paper used a structural approach in which system look at the similarities in the structure of different characters, like the location of vertical bar and the joint characters that are made up of half consonant and full character etc. The uncertainty associated with the structures of characters arises from different writing styles. To take account of this uncertainty, paper used windows to enclose certain portions of the characters for making the decisions such as whether the character is simple or joint. Another example of violation is the header line, which is not a straight line as it occupies 2 to 3 lines because of the small slant of the line. This paper given more stress on the general conditions that are applicable to most of the characters. But specific cases will be treated in the future work. Next, we will try to quantify the content of the windows to make precise decisions. The method for segmentation is working efficiently in the cases of different text sizes and different resolution images. Second method for character segmentation is also working efficiently. The method which is used for identification of middle bar, end bar, no bar character is working fine. These methods are also applicable for printed Hindi text.

REFERENCES

[1] B. M. Sagar, G. Shobha, P. Ramakanth kumar, "Character Segmentation Algorithms for Kannada optical character recognition", Proceedings of the International Conference on Wavelet Analysis and Pattern Recognition, 2008.

[2] Seong-whan Lee, Dong-June Lee, Hee-Seon Park, "A new Methodology for Gray-Scale Character Segmentation and Recognition", IEEE transactions on pattern analysis and machine intelligence, 1996.

[3] Satadal Saha, Subhadip Basu, Mita Nasipuri and Dipak Kr. Basu, "A Hough Transform based Technique for Text Segmentation", journal of computing, 2010.

[4] Y. Li, Y. Zheng, D. Doermann, and S. Jaeger, "A new algorithm for detecting text line in handwritten document", in the proceedings of International Workshop on Frontiers in Handwriting Recognition, 2006.

[5] L. Likforman-Sulem and C. Faure, "Extracting text lines in handwritten documents by perceptual grouping", Advances in handwriting and drawing: a multidisciplinary approach, C. Faure, P. Keuss, G. Lorette and A. Winter Eds, Europia, Paris, 1994.

[7] 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, 1995.

[8] C. Weliwitage, A. L. Harvey and A. B. Jennings, "Handwritten Document Offline Text Line Segmentation", in the Proceedings of Digital Imaging Computing: Techniques and Applications, 2005.

[9] A. Zahour, B. Taconet, L. Likforman-Sulem and Wafa Boussellaa, "Overlapping and multi-touching text-line segmentation by Block Covering analysis", Pattern analysis and applications, 2008.

[10] Raghuraj Singh, C. S. Yadav, Prabhat Verma, "Optical Character Recognition (OCR) for Printed Devnagari Script Using Artificial Neural Network", International Journal of Computer Science & Communication, 2010.

[11] Naresh Kumar Garg, Lakhwinder Kaur, M. K. Jindal, "Segmentation of Handwritten Hindi Text", in the International Journal of Computer Applications, (0975 – 8887), 2010. [12] Naresh Kumar Garg, Lakhwinder Kaur, MK. Jindal, "A New Method for Line Segmentation of Handwritten Hindi Text", Seventh International Conference on Information Technology, 2010.

[13] Bikash Shaw, Swapna kumar parui, malayappan, "A Segmentation Based Approach to Offline Handwritten Devanagari word Recognition.", in the International Conference on Information Technology, 2008.

[14] Supriya Deshmukh, Leena Ragha, "Analysis of Directional Features - Stroke and Contour for Handwritten Character Recognition", 2009.

[15] Bidyut B. Chaudhuri, Sumedha Bera, "Handwritten Text Line Identification in Indian Scripts", in the 10th International Conference on Document Analysis and Recognition, 2009.