Volume : 2 | Issue : 4 | April 2013 • ISSN No 2277 - 8160



Research Paper | Information Technology

A New Approach for Segmentation of Devnagari Characters

Kunal Shah	Computer Science Department, SIT, Lonavala, University of Pune
Jaideep Singh	Department of Information Technology, SIT, Lonavala, University of Pune
Prashant Pushkarna	Department of Information Technology, SIT, Lonavala, University of Pune
Hasnain Kurawadwala	Department of Information Technology, SIT, Lonavala, University of Pune
Abhishek Alate	Department of Information Technology, SIT, Lonavala, University of Pune

ABSTRACT

Optical character recognition is method that extracts comprehendible text from an image by identifying the text present. For this purpose, there is a need to divide the text into individual basic characters that can be identified by the identification mechanism of the OCR. When we talk about Devnagari script, there aren't many algorithms available that provide good segmentation of characters ready to be used by the identification mechanism. This paper presents a new approach towards segmentation of characters in an image containing Devnagari script.

KEYWORDS: Segmentation, Devnagari Handwritten Character Recognition, Optical Character Recognition

1. INTRODUCTION

Optical Character Recognition (OCR) is a technique that focuses on extracting text from an image which contains textual data. The textual data might be printed or handwritten. Handwritten Character Recognition (HCR) is a form of OCR that focuses solely on identifying or extracting textual data from an image that contains handwritten text [3]. Handwritten character recognition has become a wide research area in the past few decades. It has always been a challenging field for researchers because of variations in strokes of writers.

Extracting handwritten characters is much more complex as compared to extracting printed characters. This is due to the fact that there is hardly any uniformity in handwritten text while printed text is uniform in nature [4]. Handwritten text hugely varies depending on the person who wrote the text. There are possibilities of variations in the same document itself although it might have been written by the same person.

In this paper we propose a method of segmenting the Handwritten Devnagari Characters. Our main focus is on Sanskrit documents. In Sanskrit there are 46 basic letters which consists of 33 consonants and 13 vowels [13]. Devnagari script has its own set of problems when it comes to OCR or HCR. As a result, one of the main obstacles in designing an Optical Character Recognition (OCR) system, or specifically a Devnagari Handwritten Character Recognition (DHCR) system is segmenting the whole text into individual characters [5]. This is crucial to the system since the characters are the basic entity in the text that the system needs to identify so as to produce the handwritten text in a digital form. Improper segmentation could result in a situation where the character obtained through segmentation doesn't match any of the characters present in the character database, or it could lead to a character being recognized as a completely different one. Both of these situations lead to an increase in the error percentage in the recognition system. It is thus essential for the segmentation algorithm to be an efficient one [6-8].

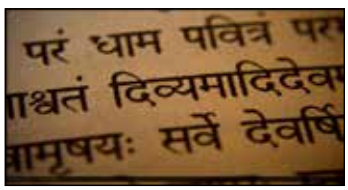

Fig. 1: Devnagari Text

2. LITERATURE REVIEW

Miss Vandana M. Ladwani and Mrs. Latesh Malik [1] proposed a system that segments modifiers in the upper zone (i.e. top modifiers), those in the lower zone (i.e. lower modifiers) and also the fused characters. This system makes use of the Neighbourhood Tracing Algorithm which helps study the pixels in the immediate neighbourhood of the current pixel under consideration. This algorithm is mainly used by this system to segment the top modifiers. The characters are then separated by finding consecutive scan lines with all pixels having the same intensity as that of the background [12]. The process of segmenting the lower modifiers involves the use of various algorithms such as Highlighted Point Detection, Vertical Line Detection and Bottom Modifier Identifier algorithms as shown in fig. 4 [11]. These algorithms help find the segmentation point between the section characters and the lower modifiers. The problem of identifying fused characters in Devnagari script is solved in this approach by making use of the average character width in the text [10]. All the characters having width greater than the average character width are deemed to be fused characters. This approach provides an accuracy of only 52% which can be improved by making certain changes to the algorithm.

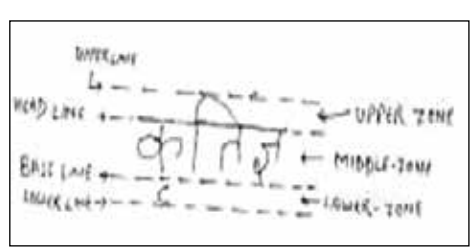

Fig. 4 Zoning

Another approach has been used by Roberto J. Rodrigues and Antonio Carlos Gay Thomé [2] in order to segment digits from an image. This approach makes use of a decision tree and Projection Profile histograms for the process of segmentation. Projection Profiles of the image along both the horizontal and vertical direction are created. After initial noise removal is done, this technique makes use of data from horizontal and vertical histograms. Successive refinements of the image are carried out using the horizontal histogram. Parts of the image that have very less corresponding height in the horizontal histogram are removed. Sub-

sequently applying this refining technique on the image helps segmenting the characters in the image. A few modifications would have to be made to this technique to make it useful for segmenting Devnagari characters since it was originally designed only to segment digits. This technique has produced very good results when tested on images containing digits.

3. PROPOSED APPROACH

This approach makes use of the Neighborhood Tracing Algorithm as well as the Projection Profile in order to provide individual characters from a word present in the text. The performance and results of the algorithm depend on the quality of the input image being provided to it. It would work best when a completely binarized image with no noise is provided to it. A binarized image is a prerequisite. However, we are not focusing on the binarization part of an image in this paper. We assume that the input image has already been binarized using one of the many algorithms available.

The algorithm first segments an individual word from the rest of the text. It begins by scanning the image from the first pixel in the top-left corner till the first black pixel is encountered. The scanning of the image is done in a left to right, top to bottom fashion. Upon encountering the first black pixel, it assumes that this pixel is a part of the word. It then takes this black pixel as the starting point and traces the boundary of the word and segments it from the rest of the image. This process carried out upto now comprises of Pass 1 of the algorithm. The algorithm in its next part eliminates the Shirorekha from the word by making use of the Projection Profile and then performs the same steps to segment individual characters from the word. The basic algorithm that forms the core for Pass 1 and Pass 2 of the proposed approach is as follows:-

Input: A binarized image after noise removal containing devnagari text.

Output: Segmented characters.

1. Scan the image left to right, top to bottom till a black pixel from the devnagari text is found. Let this pixel be A(xcur, ycur) which would be the seed pixel.

2) xmin = xcur
xmax = xcur
ymin = ycur
ymax = ycur

3) xinit = xcur
yinit = ycur

4) Let B(xn, yn) be a pixel such that
- xn = xcur – 1
- yn = ycur

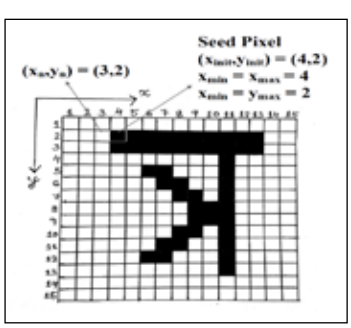

Fig. 2: Initial values for the variables used

5) IF INTENSITY(B) == 1
- B = Change_Pixel(B)

ELSE
- Move_Cur_Pixel(A, B)

6) IF xcur > xmax
- xmax = xcur

ELSE IF xcur < xmin
- xmin = xcur

7) IF ycur > ymax
- ymax = ycur

ELSE IF ycur < ymin
- ymin = ycur

8) IF xcur == xinit
- IF ycur == yinit
- Return(xmin,ymin,xmax,ymax)

9) GOTO STEP 4

In the above algorithm, the value xmin is the pixel which is a part of the current word being segmented and has the least distance from origin along x-axis of all the pixels that are a part of the word. Similarly, xmax is the pixel with the maximum distance from origin along the x-axis. ymin is the pixel which is a part of the current word being segmented and has the minimum distance from the origin along y-axis of all the pixels that are a part of the word being segmented. Similarly ymax is the pixel with the maximum distance from the origin along the y-axis. Also, INTENSITY (B) indicates the intensity of the current pixel B.
The values xmin, ymin, xmax and ymax would help segment the word. This can be done by creating boundaries at (xmin-1), (ymin-1), (xmax+1) and (ymax+1) and as a result, we obtain an isolated word. The points xmin, ymin, xmax and ymax are as shown in Fig 3. For the purpose of explanation we have only used the image of a single character rather than a complete word. The results however would be similar when the algorithm is applied to a word.

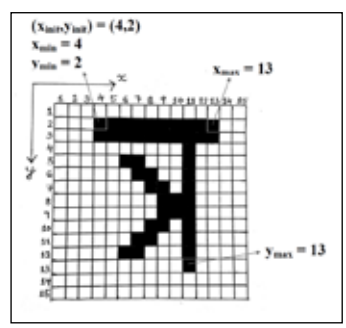

Fig. 3: Final values for the variables

After the word has been isolated, we create a Projection Profile histogram of the image containing the segmented word and use it to eliminate the shirorekha from it. This method of eliminating the shirorekha has been explained in detail in section 3.1. The same algorithm shown above is then used again on the segmented word (without the shirorekha) to obtain individual characters from it.

Once all the letters from the segmented word have been obtained, we move on to the next word by continuing with the image scanning process from the pixel position (xmax+1, ymax) and stop until the next black pixel is found. Hence, we segment all the words present in the image step by step.

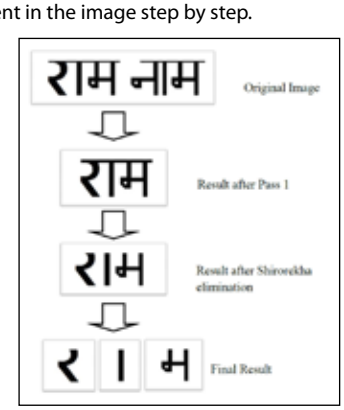

Fig. 5: Step by Step results for the proposed algorithm

In the above algorithm, the following two very important methods have been used:-

i) Change_Pixel(B)
ii) Move_Cur_Pixel(A, B)

The first method, i.e. Change_Pixel(B) is used to move the pixel B in the neighbourhood of pixel A until a black pixel is found. This is important so as to move along the boundary of the word and trace the whole boundary. This helps find the bounding values that would define the segmentation boundary for the word.

The second method, i.e. Move_Cur_Pixel (A, B) sets the current pixel A to the next black pixel found. It then also sets the pixel B to the appropriate co-ordinates for scanning the neighbourhood of the new current pixel.

Both these methods are crucial to the working of the algorithm and are very important for tracing the boundary of the word or letter that is being segmented. However, we will not be going into too much detail of how these functions actually work since they are just a series of conditional statements.

3.1 Shirorekha Elimination Using Projection Profile Histogram

A Projection Profile is a data structure used to store the number of non-background pixels in the image when the image is projected over the X-Y axes. The projection profile histogram is needed in order to obtain individual characters from the segmented and isolated word. A projection profile histogram as far as this approach is concerned is nothing but an array that would store the count of black pixels in each scan line along the x-axis.

A characteristic of Devnagari script is the Shirorekha which is a horizontal line drawn on top of all the characters present in every word. General logic thus suggests that the highest number of black pixels along the x-axis in each word would be in those scan lines that contain the Shirorekha. This logic forms the basis for this part of the algorithm. The scan line containing maximum numbers of black pixels, along with those whose count lie within the tolerance range of the maximum count, are selected. The tolerance range is identified by the tolerance value that is already set in the mechanism. This helps select all those scan lines that possibly comprise the Shirorekha. All the pixels of the selected scan lines are then set to intensity 1, i.e. white which help eliminate the Shirorekha.

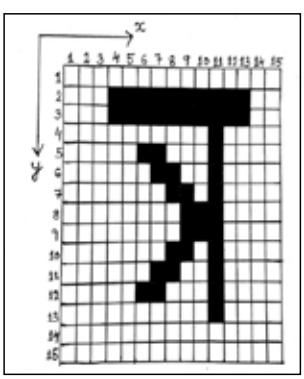

Fig. 6: Before Shirorekha elimination

Eliminating the Shirorekha from the segmented word leaves us with individual characters in the image without anything joining them. This is what helps segment these characters in the next pass of the algorithm since each of these characters could be treated as individual words again. The same algorithm mentioned above is used again on the segmented word after shirorekha removal in order to segment it further into individual characters.

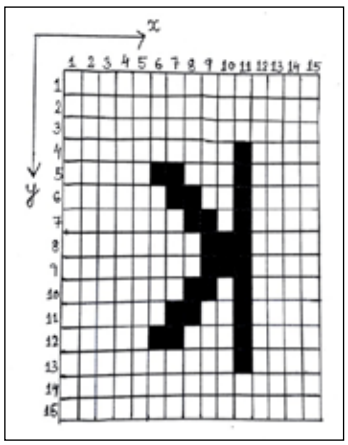

Fig. 7: After Shirorekha elimination

4. FUTURE SCOPE

The algorithm presented in this paper is a very basic one and is still in its early stages. Work can be done to improve its effectiveness in segmenting characters. It can be extended to identify fused characters in a very efficient manner or even to provide segmentation of characters from other scripts or languages. Also this algorithm gives less efficiency on segmenting words like 'BHA', 'SHA', 'KSHA' etc. With this algorithm we can work on ancient historical documents also.

5. CONCLUSION

The proposed algorithm provides a new and promising approach for segmentation of both handwritten and printed Devnagari characters. The results though depend on the quality of the image being used, with a fully binarized image containing no noise yielding the best results.

REFERENCES

[1] Vandana M. Ladwani and Latesh Malik, "Novel Approach to Segmentation of Handwritten Devnagari Word", Third International Conference on Emerging Trends in Engineering and Technology [2] Roberto J. Rodrigues and Antonio Carlos Gay Thomé, "Cursive character recognition – a character segmentation method using projection profile – based technique" [3] B. Chaudhuri, U. Pal,"OCR system to read two Indian language scripts: Bangla and Devnagari", IEEE, 1977. [4] D. Lowe, "Distinctive image features from scale invariant key points", IJCV, vol. 60, no. 2, pp. 91-110, 2004. [5] A. Bhardwaj, S. Kompalli, S. Setlur, V. Govindaraju. "An OCR based approach to word spotting in devanagari documents",SPIE ,volume 6815. 2008. [6] F. Kleber, R. Sablatnig. "Skew detection technique suitable for degraded ancient documents", CAA, 2008. [7] A. Bhardwaj, S. Setlur, V. Govindaraju, "Keyword spotting techniques for sanskrit documents", Sanskrit Computational Linguistics, Lecture Notes in Computer Science, Springer, Volume 5402, 2009, pp 403-416,Volume 5402, 2009, pp 403-416. [8] Teh C, Chin R., "On Image analysis by the methods of moments", IEEE Trans. Pattern Analysis and Machine Intelligence, 10, No. 4, pp. 496513, 1988 [9] Nikita G D Singh, "Sanskrit word recognition using Gradient feature extraction", VSRD-IJCSIT, Vol. 2(3), 2012, pp.167-174. [10] N Garg, S Kaur, "Improvement in Efficiency of Recognition of Handwritten Gurumukhi Script", IJCST Vol. 2, Issue 3, September 2011. [11] R. Jayadevan, Satish R. Kolhe, Pradeep M. Patil, and U. Pal, "Offline recognition of devanagari script: A Survey", IEEE Transaction on Systems, Man and Cybernetics-Part C: Applications and Reviews,VOL. 41, No. 6, Nov. 2011. [12] M Shahi, Dr. A K Ahlawat, B.N Pandey, "Literature survey on offline Recognition of handwritten hindi curve script using ANN approach", International Journal of Scientific and Research Publications, VOLUME 2, ISSUE 5, MAY 2012. [13] M. Frucci, G.S.D. Baja, "From segmentation to binarization of gray level image", JPAR 2008, pp 1-13