SEGMENTATION OF TEXT LINES USING SUB-IMAGE PROFILE FOR
MACHINE PRINTED TELUGU SCRIPT

J. Bharathi\textsuperscript{1}, Dr. P. Chandrasekhar Reddy\textsuperscript{2}

\textsuperscript{1}Associate Professor, Dept. of ECE, DCET, Nampally, Hyderabad, India,
\textsuperscript{2}Professor, Dept. of ECE, JNTUCE, Kukatpally, Hyderabad, India.

ABSTRACT

In thick printed documents, improperly binarized documents and manually type set documents a continuous horizontal gap in between the text lines is seldom possible, causing overlap in the horizontal profile of the two neighboring lines. The touching of the vowel modifiers and secondary form of consonants in between two consecutive text lines is extremely rare. The overlap of the line profiles of two neighboring lines renders the horizontal profile method inefficient for line segmentation. Improper segmentation of lines gives erroneous results in the connected component analysis performed for segmenting the multiple components of the single character in the next step. In the literature many researchers have analyzed this problem; however for Indian scripts especially for Telugu script this has not been fully explored. In this paper we propose to segment the lines in a page which has fewer valleys than the number of lines in horizontal profile by an algorithm that searches for a clear gap between the top and bottom lines of the inter-line space based on the statistical analysis of the overlap. The analysis of horizontal profile of a small sub-image in the segmentation path is used for selecting the correct path when two paths are possible to move further. The results of the algorithm are very encouraging with the success rate of 99.1\% on the data set considered.

Keywords: Bounding Lines, Horizontal Profile, Line Segmentation, Segmentation Path, Sub-Image Profile.

1. INTRODUCTION

OCR is the process of converting the printed document into editable form. The basic steps involved are, acquiring an image of the document, binarization of the image, extraction of characters by the segmentation process, recognition of the characters and mapping of character classes to the
standard codes for viewing. The binarization categorizes each pixel into two classes of foreground and background pixels by utilizing a property of the pixel intensity or their distribution in the image.

Extraction of characters is the main objective of segmentation. Character segmentation is an important step in the pre-processing stage, as the performance of the character segmentation can affect the performance of the overall system due to the fact that incorrectly segmented characters are not likely to be recognized properly [1]. After binarization the binarized image is segmented initially to obtain the lines of the script. The image of the line is segmented into words and then into characters. The characters are further segmented to obtain individual components.

2. LITERATURE SURVEY

One of the most popular methods used for printed documents is the Projection-profile method. The horizontal profile method is one of the effective methods used for the segmentation. The projection profile consists of the count of pixels in a horizontal scan line for all rows and plotted along a vertical line for the entire document. The white space in between the text lines is used for segmentation. This algorithm is effective when there is sufficient spacing between the two adjacent text lines. However, Telugu script consisting of vowel modifiers and consonant modifiers which extend into the white space in between two consecutive text lines will cause a problem such that the white space in the horizontal projections is not observed making the text line segmentation process prone to errors. C. V. Lakshmi and C. Patvardhan used this method in optical character recognition system for machine printed Telugu text [2].

Run Length Smearing Algorithm (RLSA) proposed by K. Y. Wong, R. G. Casey and F. M. Wahl [3] which works on binary images is used for segmenting the text line. In this algorithm the consecutive text pixels along the rows and columns are smeared. The white space in between the text pixels is filled with the black pixels if the distance is inside a predefined threshold. The bounding boxes of the connected components in the smeared image are considered as text lines. The same algorithm is used for segmenting the text lines in printed Telugu script by Negi et al., [4]. Vertical and horizontal thresholds are used to extract words. Vertical smearing is used to unite a consonant modifier with the associated base character on top of it. This smearing may join consonants and the vowel modifiers of lines that are close as shown in Fig. 1. However the adjacent lines may join when large threshold values are used. Ideal threshold values for this method are difficult to select.

![Fig. 1. Run length smeared image](image)

Distance metric between connected components for text line extraction in Telugu documents is used by Koppula et al [5]. The performance of this approach is good for closely spaced text lines and deteriorates with skew and overlapping of text lines.

Text line segmentation based on fringe maps is proposed by Negi et al. [6]. In this method each pixel of the binary image is given a fringe number. Every text pixel is given a fringe number of zero and the white pixels i.e. the background pixels is given a fringe value based on the distance from the nearest black pixel using $L_2$ metric. Fringe value information is used to segment the text lines. Initially peak fringe numbers are located. Peak fringe numbers in between the text lines are retained by filtering out the remaining. Region in between the adjacent text lines is grown based on
the peak fringe numbers and the segmentation path is found by joining the filtered PFNs. However this method needs lot of calculations for finding the fringe numbers and PFNs.

N. Priyanka et al. [7] proposed a method for segmentation of lines for Indian scripts. This method uses RLSA for smearing the text in document. The segmentation path is proposed based on vertical distances for the middle of text lines. The segmentation line is traced along the contour line of the part which obstructs the path.

Manish Kumar [8] has proposed an algorithm for the segmentation of text lines in degraded documents for Indian scripts.

3. METHODOLOGY

3.1 Segmentation of Lines

The inter-character space between the lines is characterized by a zero height valley in the horizontal profile. We can segment between the zero valued valleys to isolate each line image. However the presence of small skew in the document image, improper binarization of documents, manual typesetting used in earlier days etc., may result in the overlap of the pixel profiles in the horizontal profile (Fig. 2). The characters themselves may not have overlapped but the projection profile of top line interferes with the projection profile of bottom line thus the valley will not have a zero value. The scan line in between two text lines passes through the vowel modifiers and consonant modifiers which exist in between the two text lines. The segmentation with horizontal projection profile results in blocks of text lines (Fig. 3), lines having only consonant modifiers and lines having consonant modifiers of top line attached to the bottom line. It is observed that some font styles are more prone to touching than others.

Thus segmentation of the image of text lines having no zero valued valleys in horizontal profile method is not straightforward. In this paper we have proposed a method to segment the lines which have overlapped projection profile. This method is based on the observation that in most of the cases the characters contributing to the overlapped horizontal profile are not vertically one over the other. Therefore a segmentation path can be computed which passes through the inter-character space without touching the characters of upper and lower bounding lines. However the algorithm also effectively segments touching vowel modifiers and secondary form of consonants that occur in the segmentation path.

![Fig. 2. Overlapping projection profile](image-url)
The bounding lines of the white space in between the text lines are initially identified. A threshold for the proposed location line is fixed based on statistical analysis and moved forward to get the next point. Any ambiguities along the path are decided based on sub-image profile.

3.2. Estimating the Top and Bottom Bounding Lines of Inter-character Space

Telugu script character model has three zones viz., top zone, middle zone and bottom zone [9]. The secondary form of consonants are written in bottom zone or bottom and middle zones either to the left, right or at bottom of the base consonant. The horizontal profile is used to identify top and bottom bounding lines (Fig. 4).

\[ s_k(i) = j, \quad T_k \leq j \leq B_k \]  

Where \( s_k \) = segmentation path of \( k^{th} \) non-zero valley  
\( i = i^{th} \) column in the image  
\( j = j^{th} \) row in the image  
\( T_k \) = Top bounding line of \( k^{th} \) non-zero valley  
\( B_k \) = Bottom bounding line of \( k^{th} \) non-zero valley
Consider the horizontal profile in Fig. 5. The profiles of all the individual lines merged without any zero valued valleys. We can obtain the location of top and bottom bounding lines of the white space if we eliminate the merged portion of the profile by cutting off the lower portion of the profile and projecting the top and bottom boundaries of white space back on to the line image. The location of cutting line needs to be approximated by a statistical analysis of profile. Three cases are considered for the analysis. The lower line may not effectively remove the secondary form of consonants and vowel modifiers as only part of the profile to secondary form of consonants is removed. The middle line results in larger fragments. The identification and removing them may not yield correct results. The top line can yield bounding lines after removing the small fragments in the remaining profile. The top bounding line as indicated by any one of the three cutting lines is approximately the same as the slope of the profile is very steep and difference between the three bounding lines is negligible. However the bottom bounding line as indicated by the three lines differs from each other. This is because the number of vowel modifiers in top zone of bottom line and their shapes differ resulting in slope variation of the profile. As we move up the cutting line in horizontal profile some parts of the character may be included in the bounding space. The top two cutting lines leave few fragments of pixel run lengths corresponding to the secondary form of consonants of the top line and need to be eliminated. Based on the statistical analysis of large profiles it is found that the position of the cutting line can be considered at about 15\% of the peak value for identifying the top and bottom bounding lines of the white space. In order to avoid small lines being eliminated and consonant modifiers being attached to bottom lines, we considered the horizontal projection profile for half the width of the image.

But the small run lengths of pixels need to be removed from the horizontal profile after cutting off the lower portion of the profile as they give false boundaries. The run lengths of the pixels after eliminating the pixel values less than 15\% of the peak value are clustered into two groups. Initially the pixel run lengths of the profile are sorted in the ascending order. The minimum and maximum values of the sorted list are considered as the initial seed values of each group. A given run length is assigned to a group based on the distance between the mean value of the group and the value of run length. The group having the smaller run lengths is proposed for elimination if the highest value of the group is less than 60\% of the mean value of the second group. The starting and end location of each run length indicate the top and bottom boundaries ($T_k$, $B_k$) of white space.

### 3.3. Location of Segmentation Path

The non-zero valleys in the horizontal projection profile are occurring between the lines due to the overlap of projections of vowel modifiers of bottom line and secondary form of consonants of top line. The segmentation path should be in between the bottom line of first line and top line of second line. A mid path in the white space is a candidate for the segmentation path. However it is observed that secondary form of consonants are shifted down into the inter-line space than vowel modifiers shifted up from bottom line (Fig. 6). Hence the path also should move down. The segmenting path at 70\% of the inter-line space closely follows the contour of the characters correctly segmenting the lines.

### 3.4. Prediction of Segmentation Path

After the initial estimate of the location of the segmentation path $s_k(i)$, the next pixel of the path $s_k(i+1)$ is predicted to be at the same height as the present path. We now move up words until a pixel or the bottom line of first line whichever is encountered first. Next we move down words until a pixel or the top line of second line whichever is encountered first. The pixel at 70\% of the distance between two pixels is the point on the segmentation path (Fig. 6).

Prediction: $s_k^I(i+1) = s_k(i)$

(2)
Estimation: \( s_{k} = p_{1} + 0.7 \times (p_{1} + p_{2}) \)  \hspace{1cm} (3)

Where \( p_{1} \) = first encountered pixel either in upward direction on character or on \( T_k \)
\( p_{2} \) = first encountered pixel either in downward direction on character or on \( B_k \)

Fig. 6. Segmentation path

Fig. 7. Path 2 is the correct path with clear gap

3.5. Selection of Correct Path in Ambiguous Cases

If a pixel is encountered in the segmentation path, a decision should be made whether to move upwards or downwards to segment. We have two paths to move forward. Among the two paths in Fig. 6 the correct path is path 2. It is observed that path 1 is blocked and path 2 is open for the forward movement. A small sub-image between the top and bottom bounding lines with a width of half the average height of center to center distance of two lines is analyzed for valley in the horizontal profile. As it is most probable that secondary form of consonant is blocking the path, we look for a zero valued valley in lower half. If a valley is present we move down and move forward in between the characters. If a valley is not present we look for a valley in upper half of sub-image. If a valley is present we move up and move forward.

If neither the upward pass nor the downward pass yields a feasible path, it is concluded that the parts of top and bottom lines are touching. A side profile towards right at given location indicates a peak. The segmentation path should pass through the peak dividing the touching characters into top and bottom parts. However it is observed that, this is a rare case. Some image blocks successfully segmented are shown in Fig. 8.

Fig. 8. Successfully segmented image

4. EXPERIMENTS

Initially the horizontal profile for the document page is calculated. The horizontal profile is examined for the zero valleys and the lines are segmented along the valleys. The height of each line is found. The height of the lines are not uniform due to the presence of broken parts of characters in binarization, scanning and binarization inaccuracies and presence of more lines in a segmented
image. Each line in the Telugu script has a middle zone, top zone and bottom zone. In the study conducted on the documents in the database, it is found that only about 0.2% of lines do not have bottom zone and all the lines have top zone. The presence or absence of the zones also contributes to variation of the line height. The height of the segmented lines in a page is found to be at 40 percentile of the heights of all the segmented blocks with a confidence level of 95%.

The 40 percentile height in the sorted line heights is designated as the line height (ht\_40). The blocks having more than one line and lines having smaller height which have only consonant modifiers are found using the ht\_40.

### 4.1. Algorithm

1. Get the binarized and skew removed image
2. Construct horizontal projection profile
   \[ HPP(i) = \sum_{j=1}^{N} I(i,j) \]
3. Segment the lines
   Count the start of HPP of each line
   If \((HPP(i) = 0 \&\& HPP(i+1) = 1)\)
   \[ COUNT = COUNT + 1 \]
   If \((HPP(j) = 1 \&\& HPP(j+1) = 0)\)
   End of the line
4. Designate 40 percentile of height of the lines in the image as the line height (ht\_40)
   \[ AVG_HT = 40\% \text{ of heights of all segmented blocks} \]
5. If \(IMAGE\_BLK\_HT/AVG\_HT > 1.8\) image block has two or more lines
   \[ N\_LINE = N\_LINE + 1 \]
6. If \(IMAGE\_BLK\_HT/AVG\_HT\) between 0.3 and 0.5 the image has smaller height lines
   \[ NSM\_LINE \]
7. If \(N\_LINE\) and \(NSM\_LINE\) are zero, then exit (lines are properly segmented)
8. Construct the horizontal profile for half the width of image only to avoid smaller lines being eliminated and consonant modifiers being attached to bottom lines
   \[ HPP\_BLK(i) = \sum_{j=1}^{N/2} I\_BLK(i,j) \]
9. Segment the \(HPP\_BLK\) at 15% of max. peak height
   \[ CUT\_THRESH = 0.15*MAX(HPP\_BLK) \]
   If \((HPP\_BLK(i) < CUT\_THRESH)\)
   \[ HPP\_BLK(i) = 0 \]
10. Find the continuous pixel runs and their lengths, then sort them
    \(RL1, RL2, \ldots, RLn\)
11. Cluster the lengths into two groups of small lengths(fragments) and large lengths(characters)
    \(SEED1 = RL1\)
    \(SEED2 = RLn\)
    If \(RLi < MEAN\_GP1\),
    Assign \(RLi\) to \(GP1\)
    else
    Assign \(RLi\) to \(GP2\)
12. If \(MAX(GP1) < MEAN\_GP2\) Remove the small lengths group as they are small fragments
13. Edges of pixel runs having zero values (white space) are the top and bottom bounding lines of segmentation path
    If \((HPP\_BLK(i) = 0 \&\& HPP\_BLK(i+1) = 1)\)
14. For each inter-line space do steps 15 to 19
15. Initialize the segmentation path at 0.7*white space height
   \[ SEG_LINE(i) = 0.7 \times HT\_SPACE \]
16. Predict next segmentation point as the previous one and estimate the correct one at 0.7* the space
   \[ SEG_LINE(i+1) = SEG_LINE(i) \]
17. If the next pixel is a black one (the line is touching a part of the character) analyze a small sub image of half the height x height of inter-char space.
18. If \( SEG\_LINE(i) = 0 \) analyze sub image
   Sub image analysis
   Construct horizontal projection profile
   If valley in lower portion \( SEG\_LINE \) move down and then forward
   If valley in upper portion \( SEG\_LINE \) move up and then forward
   Else the characters are touching, construct side profile on left towards right, segment through peak
19. Complete the segmentation path till the end of image.

\[ Fig. 9. \text{Kannada script segmentation with horizontal profile} \]

\[ Fig. 10. \text{Segmentation with proposed algorithm} \]


5. RESULTS

About 148 pages of 10 different books binarized and de-skewed are considered for the analysis which also includes binarized documents downloaded from Digital Library of India (DLI). The segmented lines using the horizontal profile have one, two or more text lines. Each line block is segmented using the proposed algorithm. The results are shown in Table I.

<table>
<thead>
<tr>
<th></th>
<th>Projection profile method</th>
<th>Proposed method</th>
</tr>
</thead>
<tbody>
<tr>
<td>No of Pages analyzed</td>
<td>148</td>
<td>148</td>
</tr>
<tr>
<td>Total number of lines</td>
<td>4008</td>
<td>4008</td>
</tr>
<tr>
<td>No of blocks having multiple text lines</td>
<td>612</td>
<td>0</td>
</tr>
<tr>
<td>No of lines correctly segmented</td>
<td>2559</td>
<td>3972</td>
</tr>
<tr>
<td>Accuracy</td>
<td>63.8%</td>
<td>99.1%</td>
</tr>
</tbody>
</table>
All the images have overlapping projection profiles though the characters are not touching. The failure cases resulted due to unusual separating space between the base character and secondary form of conjunct consonants, characters not aligned horizontally along a line and broken characters in the documents. The algorithm is also tested to segment Kannada and Hindi scripts which have given correct results for the few documents tested.

6. CONCLUSION

The horizontal profile method for text line segmentation is not effective when the zero valued valleys are not present due to overlapping of text line projections. The proposed algorithm segments the document page effectively using the clear gap in between the text lines and sub-image profile valleys in cases of ambiguity. We can avoid smaller lines being eliminated by considering only half the width of the document image for calculating horizontal projection profile. The algorithm worked well for Hindi and Kannada documents.

REFERENCES


190
AUTHORS’ PROFILE

J. Bharathi received her B.Tech degree in Electronics and Communication Engineering from Acharya Nagarjuna University, Guntur, India. She received her M.Tech degree in Digital Systems and Computer Electronics from Jawaharlal Nehru Technological University, Hyderabad, India. She joined as faculty member in Electronics and Communication Engineering Department, Deccan College of Engineering and Technology, Hyderabad, India and is currently working as Associate Professor. Her research interests include Image Processing, Speech and Signal Processing.

Dr. P. Chandrasekhar Reddy received his B.Tech. degree in Electronics and Communication Engineering from Jawaharlal Nehru Technological University, Hyderabad, India and M.E. from Bharatiyar University, Coimbatore. He received his M.Tech and Ph.D from Jawaharlal Nehru Technological University, Hyderabad, India. He joined as faculty in JNTU, Anantapur. Currently he is working as Professor Co-ordination in JNTU, Hyderabad, India. He is an author of numerous technical papers in the Fields of High Speed Networking and Wireless Networks. His research interests include Mobile and Wireless Communications and Networks, Personal Communications Service and High Speed Communications and Protocols.