Line and Word Segmentation Approach for Printed Documents

Nallapareddy Priyanka
Computer Vision and Pattern Recognition Unit
Indian Statistical Institute,
203 B.T. Road, Kolkata-700108, India

Srikanta Pal
Computer Vision and Pattern Recognition Unit
Indian Statistical Institute,
203 B.T. Road, Kolkata-700108, India

Ranju Mandal
Computer Vision and Pattern Recognition Unit
Indian Statistical Institute,
203 B.T. Road, Kolkata-700108, India

ABSTRACT
Line and word segmentation is one of the important step of OCR systems. In this paper we have proposed a robust method for segmentation of individual text lines based on the modified histogram obtained from run length based smearing. A complete line and word segmentation system for some popular Indian printed languages is presented here. Both foreground and background information are used here for accurate line segmentation. There may be some touching or overlapping characters between two consecutive text lines and most of the line segmentation errors are generated due to touching and overlapping character occurrences. Sometimes, interline space and noises make line segmentation a difficult task. Our method can take care of this situation accurately. Word segmentation from individual lines is also discussed here. We have tested our method on documents of Bangla, Devnagari, Kannada, Telugu scripts as well as some multi-script documents and we have obtained encouraging results from our proposed technique.

General Terms
Line segmentation, Word segmentation, Histogram, Indian documents

Keywords
Optical Character Recognition, Document Analysis, Multi-scripts document

1. INTRODUCTION

The objective of Optical Character Recognition (OCR) is automatic reading of optically sensed document text materials to translate human-readable characters to machine-readable codes. Research in OCR is popular for its various application potentials in banks, library automation post-offices and defense organizations. Other applications involve reading aid for the blind, library automation, language processing and multi-media design. Character recognition as an aid to the visually handicapped was at first attempted by the Russian scientist Tyurin in 1900. The OCR technology took a major turn in the middle of 1950s with the development of digital computer and improved scanning devices. For the first time OCR was realized as a data processing approach, with particular applications for the business world. From that perspective, David Shepard, founder of the Intelligent Machine Research Co. can be considered as a pioneer of the development of commercial OCR equipment. Currently, PC-based systems are commercially available to read printed documents of single font with very high accuracy and documents of multiple fonts with reasonable accuracy. In Optical Character Recognition (OCR), the text lines in a document must be segmented properly before recognition. Correctness/incorrectness of text line segmentation directly affects accuracies of word/character segmentation and consequently changes the accuracies of word/character recognitions. Several techniques for text line segmentation are reported in the literature [3, 5, 8, 11, 13, 15]. These techniques may be categorized into three groups as follows: (i) Projection profile based techniques, (ii) Hough transform based techniques, (iii) Thinning based approach.

As a conventional technique for text line segmentation, global horizontal projection analysis of black pixels has been utilized in [1, 2, 13, 15]. Partial or piece-wise horizontal projection analysis of black pixels as modified global projection technique is employed by many researchers to segment text pages of different languages [3, 4,16]. In piece-wise horizontal projection technique text-page image is decomposed into vertical stripes. The positions of potential piece-wise separating lines are obtained for each stripe using partial horizontal projection on each stripe. The potential separating lines are then connected to achieve complete separating lines for all respective text lines located in the text-page image.

Concept of the Hough transform is employed in the field of document analysis in many research areas as skew detection, slant detection, text line segmentation, etc [10]. Based on Hough peaks, text lines can be separated and many pieces of earlier work can be available on it.

Thinning operation also is used by researchers for text line segmentation from documents [12]. In [12], a thinning algorithm followed by post-processing operations is employed in the entire background region to detect the separating borderlines. Since the thinning algorithm is applied on entire background of text page to obtain separation lines, it requires post-processing to remove extra branches.

In this paper we have proposed a robust method for segmentation of documents info lines and words and the method is based on the modified histogram obtained from run length based smearing. Foreground and background information is also used for accurate line segmentation. There may be some touching or overlapping characters between two consecutive text lines and most of the line segmentation errors are generated due to this. Our method can take care of these situations accurately.

The organization of the rest of this paper is as follows: In Section 2, we have discussed properties of Indian scripts considered here. Section 3 details proposed approach. Experimental results and performance analysis are discussed in Section 4. Finally in section 5, the paper is concluded.

2. PROPERTIES OF INDIAN SCRIPTS USED HERE

There are twelve scripts in India and in most of these scripts the number of alphabets (basic and compound characters) is more than two hundred fifty.

Most of the Indian scripts are originated from Brahmi script through various transformations. Writing style of the Indian scripts considered in this paper is from left to right, and concept of upper/lower case is absent in these scripts. Among Indian scripts, Devnagari is the most popular script in India and the most popular Indian language Hindi is written in Devnagari script. Nepali, Sanskrit and Marathi are also written in Devnagari script. Moreover, Hindi is the national language of India and the third most popular language in the world [1]. Devnagari script has 52 symbols (10 vowels, 2 modifiers and 40 consonants). Alphabets are known as "matra" symbols. Matra symbols are used when consonants and vowels are to be written together.

Bangla, the second most popular language in India and the fifth most popular language in the world, is an ancient Indo-Aryans language. Bangla script alphabet is used in texts of Bangla, Assamese and Manipuri languages. Bangla is also the national language of Bangladesh. Also Bangla is the official language of West Bengal State of India. Bangla is an Indo-Aryan language with around 211 million speakers in Bangladesh, the Indian state of West Bengal. Alphabet set of this script has 11 vowels, 40 consonents, and 10 numerals called basic characters. There are also more than 200 compound characters which is formed by combination of two or more basic characters.

Telugu is the 3rd most popular scripts in India. It is the official language of the southern Indian state, Andhra Pradesh. Telugu is also spoken in Bahrain, Fiji, Malaysia, Mauritius, Singapore and the UAE. The Telugu script is closely related to the Kannada script. It's a Dravidian language spoken by about 75 million people. The script has 16 vowels and 36 consonants.

Kannada is a popular script and it is the official language of the southern Indian state, Karnataka. Kannada is a Dravidian language mainly used by the people of Karnataka, Andhra Pradesh, Tamil Nadu and Maharashtra. Kannada is a Dravidian language spoken by about 44 million people. The language has 52 characters in its alphabet set. Fig.1-8 shows the alphabet sets of Bangla, Devnagari, Telugu and Kannada scripts.

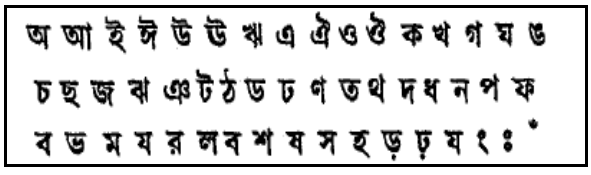


Fig.1. Basic characters (First 11 are Vowels and rest are Consonants) of Bangla Script

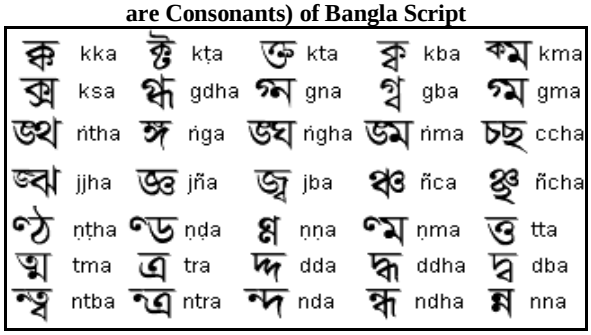


Fig.2. Some Bangla compound characters

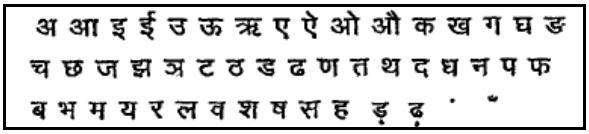


Fig.3. Basic characters of Devnagari scripts (First 11 are vowels and rest are consonants)

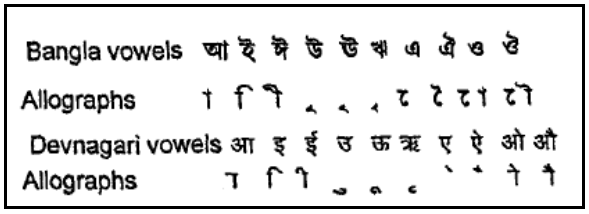


Fig.4. Vowels modifiers of Bangla and Devnagari

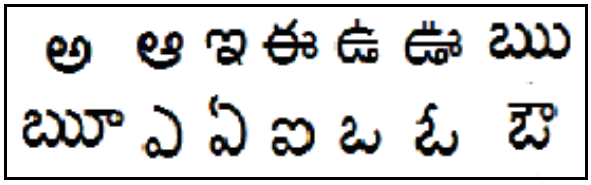


Fig.5. Vowel of Telugu Script

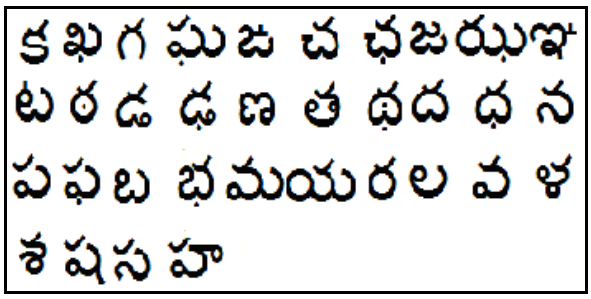


Fig.6. Consonants of Telugu Script

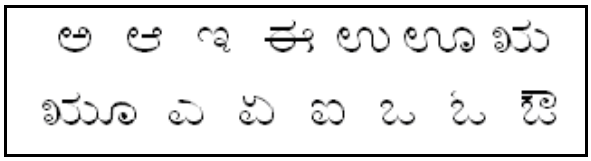


Fig.7. Vowels of Kannada Script

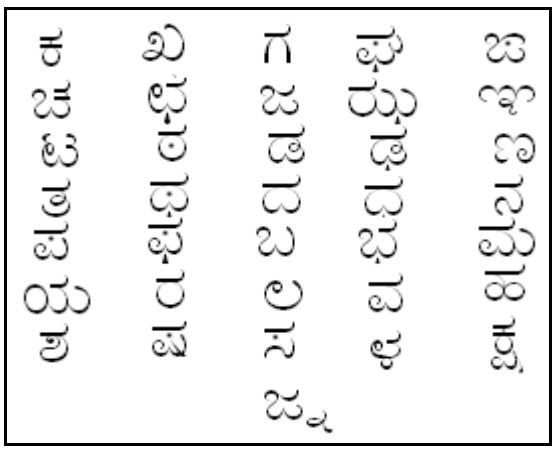


Fig.8. Consonants of Kannada Script

3. PROPOSED APPROACH

We have collected the printed document pages from different books, newspapers and magazines. The document pages are scanned using a flat bed scanner at a resolution of 300 dpi and stored as gray scale image in tiff format. Before the actual line and word segmentation step, the input image must be pre-processed; the first step is binarization, and the second step is skew detection and correction. Then the lines and words can be segmented properly.

3.1 Binarization

Document image binarization is an useful method to convert a gray image into two tone. Global binarization and locally adaptive binarization are two popular types of binarization methods. There are few categories of binarization methods, such as Histogram-based, Clustering-based, Entropy-based, Object attribute-based, Spatial binarization and Locally adaptive etc. Histogram based method use the properties of histogram, such as peaks and valleys or concavities. Clustering based methods use partitioning the image pixels into two clusters to find the threshold value. Entropy based methods use entropy information. Object attribute based methods use attribute of image like edge matching or measure the similarity between the original and the binarized image. Spatial based methods mainly use higher order probability or the correlation between pixels and locally adaptive based methods compute local threshold based on the information contained in the neighborhood of each pixel. We have used histogram based properties to binarize the documents taken as a data set. The digitized text images are first converted into two-tone images using a histogram based thresholding approach. Here we represent object pixels by 1 and background pixels by 0. The two-tone image generally shows protrusions and dents in the characters as well as isolated object pixels over the backgrounds, which are cleaned by a logical smoothing approach [2].

3.2 Skew estimation and correction

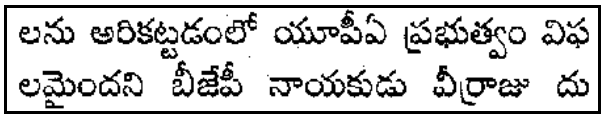
When a document is fed to the optical sensor either (scanner) mechanically or by a human operator to get the digital image, a few degrees of skew (tilt) is unavoidable. Skew angle is the angle that the text lines in the digital image makes with the horizontal direction. Skew estimation and correction are important preprocessing steps of line and word segmentation approaches. It deals with skew estimation of a class of scripts that includes some major Indian Languages like Bangla, Devnagari, Kannada, and Telugu. Skew correction can be achieved by (i) estimating the skew angle, and (ii) rotating the image by the skew angle in the opposite direction. In this work, we use a Hough transform based technique for skew angle estimation. To reduce the amount of data to be processed by the Hough transform, we compute some candidate points considering some selected components from the image. Because of the use of this selected components small and irrelevant components like dots, punctuation marks, small modifiers, etc. are mostly filtered out of the skew estimation process. We detect the topmost and bottom most points of the components and we this points for skew detection.

3.3. Line segmentation

There are several steps in the line segmentation method that are systematically described below.

Step-1: Run length smearing

A smoothing algorithm is applied in the text of a document page. In this step we use run length smearing technique [11] to increase the strength of the histogram. Here we consider the consecutive run of white pixels in between two black pixels and we compute the length of that white run. If the length of white run is less than five times of stoke width (thickness of a line in a font character), we fill up the white run length into black. In Fig.9(a) there are two original text lines and in Fig.9(b) there are smoothed text lines with horizontal histogram corresponding of their two text lines.



(a)

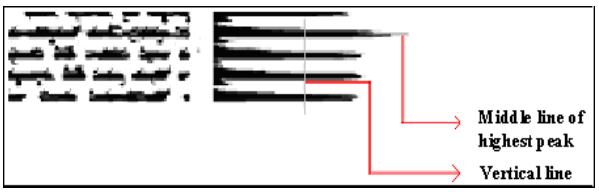
(b)

Fig.9(a) Original text lines (b) Smoothed text lines with histogram.

Step-2: Recursive procedure to get middle lines for segmentation.

Getting the histogram of every line from the smoothed document page, we consider the highest peak among all the peaks of the horizontal projection profile. After that we find the middle point of length of highest peak, and then we draw a vertical line from top to bottom at the middle point of the highest peak as shown in Fig.10(a).

The continuity of this step is to find the middle lines of each and every peaks of histogram. At the line (the line passes vertically through middle point of the highest peak) we find middle point of peaks. We draw the horizontal lines based on this middle point of the width of histogram. In some cases all peak of histograms do not cross this vertical line. For these cases we find distances between middle lines and find the average value of these distances. If the distance between the two middle lines is greater than two times of average value then we assume that region contains one or more text lines and we need recursive segmentation for that region. After getting that region (the region between two middle lines of peaks) we apply the same procedure to find vertical line through the middle of highest peak and middle lines of that particular region. This procedure runs recursively; until we find middle lines of particular image as shown in Fig .10(b).



(a)

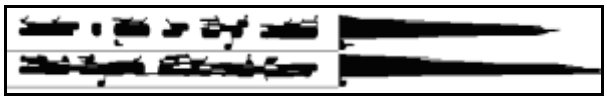
(b)

Fig. 10(a) Highest peak and vertical line drawn at the middle of highest peak is shown in a Telugu script document (b) Middle line detection for considering small length text line (shown in Telugu).

Step-3: Finding candidate line

In this step, from the starting point of first histogram we vertically scan the region in between the first middle and second middle line of histogram until we get first two white pixels. We consider that two white pixels as minimum points. The line, where we get the first white pixel, we consider that line as first minimum. Similarly the line where we get second white pixel, we consider that line as second minimum. Now we calculate the vertical distances from first middle line to first minimum point and from first middle line to second minimum point. Getting these two distances, we consider the maximum distance. The minimum point which contains maximum vertical distance as a separator between two consecutive middle lines. In this way we find all line separators between two consecutive middle lines and shown in Fig. 11(a) and Fig .11(b). If we consider only the point where we get minimum black pixel in the histogram is separator line, then we will get many errors. To remove such errors we use the above technique for candidate line separator and this technique is useful for all scripts like Bangla, Devnagari, Telugu and Kannada scripts considered here.

(a)



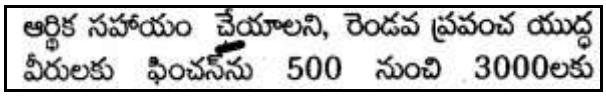
(b)

Fig.11(a) Initial segmentation line through the white pixels of horizontal histogram (b) Result after considering only the candidate lines from the initial line segmentation.

Step-4: Resolving the problems of overlapping and touching component

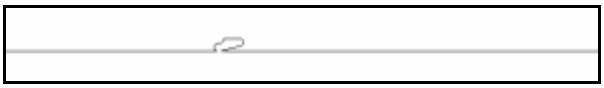
In a text-page, either overlapping or touching or both the problems of overlapping and touching may occur in many positions of two consecutive text lines in the text-page. Therefore, the problem of text line segmentation cannot be completed without treating the overlapping and touching cases. In the present technique subsequent to getting separating lines, it should be checked whether each separating line passes through the white gap between two consecutive lines or it crosses some components of text lines. To do so, each separating line is traced from left to right and as soon as the separating line passes through black pixels of a component in the text-page, a problem of touching/overlapping happens. If a separating line does not pass through any black pixel of any component then neither touching nor overlapping occurs in that separating line. In order to check whether an overlapping or a touching has occurred, the height of the component having intersection with separating line is examined. If the height of the component, which having intersection with separating line, is greater than average height of components present in the input text-page, the component is judged as a touching component. Otherwise, it is considered as an overlapping component.

To take care of the problem of overlapping, the contour points of the component are traced. The intersection point of the separating line and the component is considered as starting point for contour traversal. The direction of contour traversal (upwards or downwards) is selected based on number of contour pixels, which lie in upper or lower parts based on the position of starting point. If the number of contour pixels in upper side of the starting point is smaller (greater) than the number of contour pixels in lower side, the upper (lower) contour pixels are considered. An (A) upwards (downwards) traversing is continued until it comes back to the same row of starting point after reaching the top (bottom) point of the component. Thus if an overlapping occurs, the separating line follows the contour of component so that the overlapping portion can be included in the respective text line based on contour tracing. Fig.12 illustrates the overlapping problem.



(a)

(b)

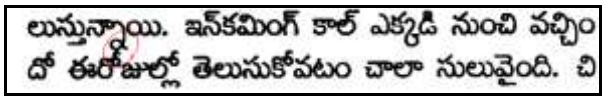


(c)

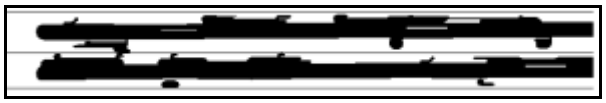
Fig.12 Illustration for overlapping (a) Part of Input image with overlapping characters (b) Smoothed image with overlapping characters (c) Movement of separator line for text line separation is shown.

For touching component segmentation, at first we fix a touching zone with height of T_1, which is defined as T_1 = (statistical mode of heights of white spaces obtained from background between two consecutive lines in the text image)/2. Touching generally occurs in this zone. Based on a statistical study, it was explored that when two components touched, in most of the cases the positions with minimum stroke width within this zone happened to be the position of touching of two components.

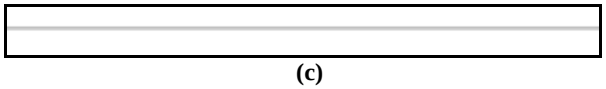
Therefore, for touching segmentation a separating line crosses a touching component through the position with minimum width in the defined touching zone. Searching for minimum stroke width is vertically performed from the position of the intersection point upwards and downwards each with height of $T_1/2$. Illustration of touching are shown in Fig.13



(a)



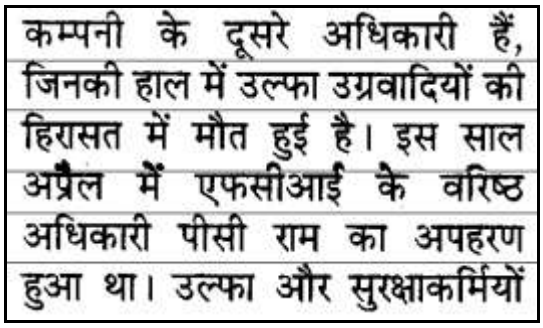
(b)



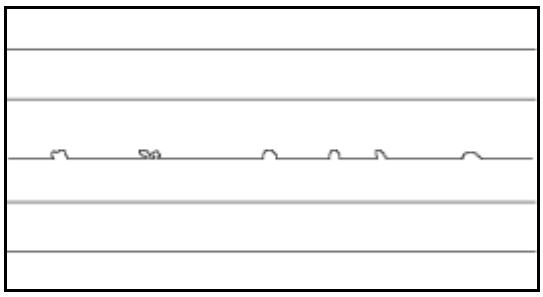
(c)

Fig.13 Illustration of touching (a) Part of Input text with touching letters (b) Smoothed image (c) Movement of separator for line separation.

(a)



(b)



(c)

Fig. 14 (a) An original Hindi document (b) Line separators between the lines (c) Only line separators.

3.4 Word segmentation

In word segmentation method, a text line has taken as an input. After a text line is segmented, it is scanned vertically. If in one vertical scan two or less black pixels are encountered then the scan is denoted by 0, else the scan is denoted by the number of black pixels. In this way a vertical projection profile is constructed. Now, if in the profile there exist a run of at least $k1$ consecutive 0's then the midpoint of that run is considered as the boundary of a word. The value of $k1$ is taken as 1/3 of the text line height. Word segmentation results of a Telugu text line are shown in Fig.15.

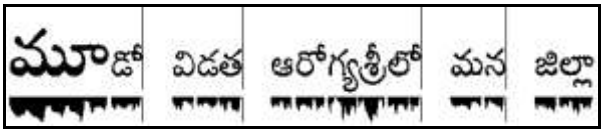


Fig.15 Output for word segmentation

4. RESULT AND DISCUSSION

4.1 Data set

We tested our algorithm on 1108 lines in Bangla, 1022 lines in Hindi, 1010 lines in Telugu,1007 lines in Kannada and 1254 lines in Multilingual documents. These documents images are taken from different types of books, newspaper, magazines etc. Also the texts are of different fonts.

4.2 Results on single script

The detail results of line segmentation are shown in Table 1 and Table 2 respectively. From the tables it can be seen that in all the scripts our proposed method give more than 99% accuracy. We tested our algorithm on 4147 lines of single script documents and we got overall 99.5% accuracy.

Table 1. Text line separation accuracy

Languages	No of lines in the documents	Line segmentation accuracy
Bangla	1108	99.54%
Hindi	1022	99.41%
Telugu	1010	99.10%
Kannada	1007	99.70%
Total	4147	99.5% (Overall)

4.3 Results on multi-script data

We also tested our algorithm on bi/multi-script documents. We considered 379 text lines of Bangla and Hindi bi-lingual documents, 278 text lines of Hindi and English bi-lingual documents, 324 text lines of Telugu and English bi-lingual documents, 255 text lines of Kannada and English bi-lingual documents and the accuracy we obtained 99.63%, 99.55%, 99.34 %, 99.58%, respectively. The results computed on bi-lingual documents are given in table 2. We obtained an overall accuracy of 99.5% from the multi-script documents.

Table 2. Multi-script text line separation accuracy

Multi-script documents	No of lines in the documents	Line segmentation accuracy
Bangla & Hindi	397	99.63%
Hindi & English	278	99.55%
Telugu & English	324	99.34%
Kannada & English	255	99.58%
Total	1254	99.5% (Overall)

4.4 Word segment results

We have got an encouraging result on word segmentation also. The obtained result of word segmentation is given in Table 3. We have tested the algorithm on single script and multi-script documents and the documents contain 9966 words. The overall word segmentation accuracy we have got from the experiment is 99.55 %. Word segmentation results of single script and multi-script data are given in Table 3.

Table 3. Word Separation result

Languages	No of words	Accuracy in word segmentation
Bangla	1560	99.50%
Hindi	1925	99.47%
Telugu	2006	99.65%
Kannada	1745	99.54%
Multi-lingual Document	2730	99.60%

4.5 Comparison of results

Kumar et al. [13] described a line segmentation method based on Horizontal and Vertical projection .They have used only 10 printed documents as an experimental data set. They have not mentioned any algorithm for skewed image and also for overlapping and touching component. The data set contains only 156 lines of data and 886 words which is not an adequate amount for an experiment. Their work has claimed 100% accuracy in line segmentation and more than 99% accuracy in word segmentation.

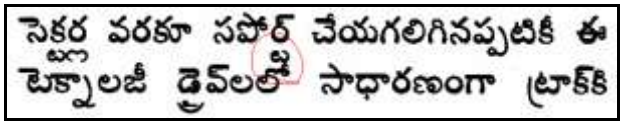
Jindal et al.[14] considered 10 different types of strips of different languages like Gurumukhi, Devnagari, Bangla and Gujrati for their experiment. The algorithm designed for segmenting horizontally overlapping and touching lines. The volume of data they have used is not clearly mentioned in this paper. They have got an accuracy of 96.45-99.79% on overlapping lines in the different sized text in printed newspapers in Gurumukhi script.

Table 4. Comparison of results with other methods

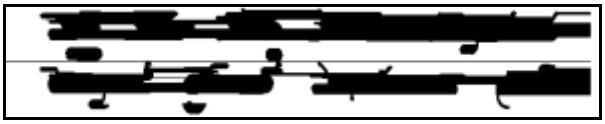
Method Proposed by	Method based on	Data set used	Accuracy on line segmentation	Accuracy on word segmentation
Vijay et.al[13]	Horizontal and vertical projection	156 lines and 886 words of Hindi and Gurumukhi scripts.	100%	99.75 %
Jindal et al.[14]	Projection based method	50-100 news items of each script	96.45-99.79%	Nil
Proposed method	Horizontal and vertical projection, Recursive middle line extraction, Component contour tracing	5158 lines and 9966 words of Bangla, Hindi, Telugu, Kannada and Multilingual scripts.	99.5%	99.54 %

4.6 Erroneous Results

From the experiment we noted that most of the errors in line segmentation are due to single character which is isolated in fashion. In the documents scripts where some of the characters lies in the lower or upper part of a characters, we noted that our proposed method generate errors due to these sort of characters. Because of such positional information or characters sometimes a character of a line touches to the characters to its lower or upper line. Example of an error in Telugu script is shown in Fig.16. Here, the error is due to the touching character, marked in Fig.16(a) by circle.



(a)



(b)

Fig.16(a) Input image with touching characters (b) Smoothed image with line separators. Here error is obtained in line separation due to a component.

5. CONCLUSION

Line and word segmentation is one of the important steps of OCR systems. In Optical Character Recognition (OCR), the text lines in a document must be segmented properly before recognition. In this paper we have proposed a robust method for segmentation of individual text lines based on the modified histogram obtained from run length based smearing. Both foreground and background information are used here for taking care of touching characters for accurate segmentation. From the experiment of Bangla, Devnagari, Kannada, Telugu scripts as well as some multi-script documents, and we obtained encouraging results from our proposed technique.

6. REFERENCES

1. U. Pal and B.B. Chaudhuri, "Indian script character recognition: A Survey", Pattern Recognition, vol. 37, pp. 1887-1899, 2004.
2. B. B. Chaudhuri and U. Pal, "A complete printed Bangla OCR system", Pattern Recognition, vol.31, pp.531-549, 1998.
3. K. Wong, R. Casey and F. Wahl "Document Analysis System ", IBM j.Res . Dev., 26(6), pp.647-656, 1982.
4. Likforman-Sulem, L., Zahour, A. and Taconet, B., "Text line Segmentation of Historical Documents: a Survey", International Journal on Document Analysis and Recognition, Springer, Vol. 9, Issue 2, pp.123-138, 2007.
5. F. Hones and J. Litcher, "Layout extraction of mixed mode documents", Machine Vision Application, vol. 7, pp. 237–246, 1994.
6. K. Kise, W. Iwata, and K. Matsumoto, "A computational geometric approach to text line extraction from binary document images", in Proc. IAPR Workshop Document Analysis Systems, pp. 364-375, 1998.

7. D. S. Le, G. R. Thoma, and H. Wechsler, "Automatic page orientation and skew angle detection for binary document images", Pattern Recognition, vol. 27, pp. 1325-1344, 1994.
8. G. Nagy, S. Seth, and M. Viswanathan, "A prototype document image analysis system for technical journals", Computer, vol. 25, pp. 10-22, 1992.
9. L. O'Gorman, "The document spectrum for page layout analysis", IEEE Trans. Pattern Anal. Mach. Intell., vol. 15, pp. 1162–1173, 1993.
10. U. Pal, M. Mitra, and B. B. Chaudhuri, "Multi-skew detection of Indian script documents", in Proc. 6th Int. Conf. Document Analysis Recognition, pp. 292-296, 2001.
11. H. Yan, "Skew correction of document images using interline cross-correlation", CVGIP: Graph. Models Image Process, vol. 55, pp. 538–543, 1993.
12. G. Magy, Twenty years of Document Analysis in PAMI, IEEE Trans. In PAMI, Vol.22, pp. 38-61, 2000.
13. Vijay Kumar, Pankaj K.Senegar, "Segmentation of Printed Text in Devnagari Script and Gurmukhi Script ", IJCA: International Journal of Computer Applications, Vol.3,pp. 24-29, 2010.
14. M.K. Jindal, R.K. Sharma and G.S. Lehal,"Segmentation of Horizontally overlapping Lines in Printed Indian Scripts",International Journal of Computational Intelligence Research,vol-3, pp.277-286, 2007.
15. U. Pal and Sagarika Datta, "Segmentation of Bangla Unconstrained Handwritten Text", Proc. 7th Int. Conf. on Document Analysis and Recognition, pp.1128-1132, 2003.
16. U. Pal and P. P. Roy, "Multi-oriented and curved text lines extraction from Indian documents", IEEE Trans. On Systems, Man and Cybernetics- Part B, vol.34, pp.1676-1684, 2004.