Global Journal of Computer Science and Technology
Graphics & Vision
Volume 13 Issue 5 Version 1.0 Year 2013
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Character Segmentation for Telugu Image Document using Multiple Histogram Projections

By N. Anupama, Ch. Rupa & Prof. E. Sreenivasa Reddy
Acharya Nagarjuna University, India

Abstract - TEXT line segmentation is one of the major component of document image analysis. Text line segmentation is necessary to detect all text regions in the document image. In this paper we propose an algorithm based on multiple histogram projections using morphological operators to extract features of the image. Horizontal projection is performed on the text image, and then line segments are identified by the peaks in the horizontal projection. Threshold is applied to divide the text image into segments. False lines are eliminated using another threshold. Vertical histogram projections are used for the line segments and decomposed into words using threshold and further decomposed to characters. This approach provides best performance based on the experimental results such as Detection rate DR (98%) and Recognition Accuracy RA (98%).

Keywords : *optical character recognition, segmentation, histogram projection, telugu scripts.*

GJCST-F Classification: *I.4.6*

Strictly as per the compliance and regulations of:

Character Segmentation for Telugu Image Document using Multiple Histogram Projections

N. Anupama [α], Ch. Rupa [σ] & Prof. E. Sreenivasa Reddy [ρ]

Abstract - TEXT line segmentation is one of the major component of document image analysis. Text line segmentation is necessary to detect all text regions in the document image. In this paper we propose an algorithm based on multiple histogram projections using morphological operators to extract features of the image. Horizontal projection is performed on the text image, and then line segments are identified by the peaks in the horizontal projection. Threshold is applied to divide the text image into segments. False lines are eliminated using another threshold. Vertical histogram projections are used for the line segments and decomposed into words using threshold and further decomposed to characters. This approach provides best performance based on the experimental results such as Detection rate DR (98%) and Recognition Accuracy RA (98%).

Keywords : optical character recognition, segmentation, histogram projection, telugu scripts.

I. Introduction

Text line segmentation is an essential pre-processing stage for recognition in many Optical Character Recognition (OCR) systems. Segmentation of text line is a vital step because inaccurately segmented text lines result in errors during recognition stage. Segmentation of the handwritten document is still one of the most concerned challenging problems. Several techniques for text line segmentation are reported in the literature for segmenting Indian script documents. These methods include projection profile (white space analysis) [1], voronoi and docstrum [2], graph cut, connected components based. Segmentation is not accurate with these methods. Jawahar [3] proposed the graph cut method that requires a priori information about the script structure to cut. Rajasekharan proposed a method based on projection method for Kannada script document segmentation [4]. As a conventional technique for text line segmentation, global horizontal projection analysis of black pixels has been utilized in [5, 6, 7, 8]. 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 [9, 10, 11]. In piecewise horizontal projection technique text-page image is decomposed into vertical strips. The positions of potential piece-wise separating lines are obtained for each strip 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.

In this paper a robust method for segmentation of documents into lines and words and the proposed method is based on the modified histogram as the Telugu script is very complex. For accurate line segmentation Foreground and background information is also used. This method take cares of eliminating false lines and recovering the loss of text in overlapped text lines.

The rest of the paper is organized as follows: In Section 2, we discussed the properties of Telugu scripts considered here. Proposed approach is discussed in Section 3. Experimental results in Section 4. Finally the paper is concluded in section 5.

II. Characteristics of Telugu Script

Telugu is the most popular South Indian spoken script based language. The Telugu character set contains 16 vowels, 36 consonants, vowel (maatras) and consonant modifiers (vaththus). These characters are combined to represent several frequently used syllables (estimated between 5000 and 10000) in the language [12, 13, 14]. We refer to these basic orthographic units as glyphs (single connected component representation). These characters will have variable size. (i.e. width and height). In Latin based scripts most of the characters have same size except few characters. Segmentation of such characters is difficult when compared with Latin based scripts like English. The figure 1 shows sample Telugu simple and compound character images.

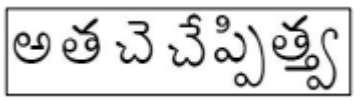


Figure 1 : Examples for simple and compound characters

III. Proposed Approach

Here we propose a new technique which automatically identify and segment the text line regions of handwritten documents. Figure 2 shows the basic steps in our proposed algorithm.

Authors α σ : CSE Department Acharya Nagarjuna University.
E-mails : namburianupama@gmail.com, edara_67@gmail.com
Author ρ : CSE Department VVIT, Nambur.
E-mail : v12.balaji@yahoo.com

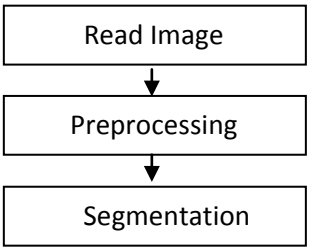


Figure 2 : Shows the basic steps in segmentation algorithm

a) Pre Processing

The raw data is subjected to a number of preliminary processing steps to make it usable in the stages of character analysis.

Pre-processing aims to produce data that are easy for segmentation accurately. The main objectives of pre-processing include:

Binarization
Noise reduction
Skeletonization/Normalization
Skew correction.

We have used binary image for our work and to convert the original grey-level document images into binary image, we have applied the algorithm due to Otsu [15]. Then noise removed, skew corrected output image from the pre-processing phase is given as input to the Segmentation stage. For Noise removal we use morphological operators. Figure 3 shows steps in Noise removal.

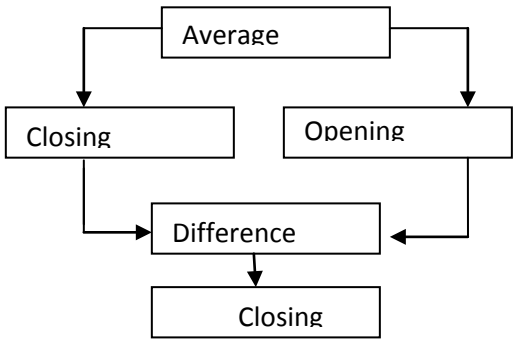


Figure 3 : Steps in Noise Removal

$$E(I(x,y) = \frac{1}{mn} \sum_{i=-\frac{m}{2}}^{i=\frac{m}{2}} \sum_{j=-\frac{n}{2}}^{j=\frac{n}{2}} I(x+i, y+j) \quad S(i,j) \qquad (1)$$

$$I(x,y).S = (I(x,y) \oplus S) \odot S \qquad (2)$$

$$I(x,y) \circ S = (I(x,y) \odot S) \oplus S \qquad (3)$$

$$D(x,y) = I1(x,y) - I2(x,y) \qquad (4)$$

$$T(I(x,y)) = \begin{cases} 255, if\ I(x,y) > T \\ 0\ otherwise \end{cases} \qquad (5)$$

In Equations 1-5

I(x, y) denotes the gray level value of the pixel located at position (x, y).

S is the structural element of size m x n where m and n are odd values larger than zero. Here we use a structuring element of 3x3 in mathematical morphological contect. Closing operation is performed to turn the border of the resulting image more compact and closer. Normalization provides a tremendous reduction in data size, thinning extracts the shape information of the characters. The document then has to be skew corrected which is the input for Segmentation.

b) Segmentation

Once the pre-processing is completed then the histogram projections in y direction are obtained in order to perform Line segmentation and then x histogram projections for words and character segmentation.

$$Profile(y) = \sum f(x,y)$$

i. *Line segmentation*

It is the process of identifying lines in a given image.

Steps for the line Segmentation is as follows

1. Scan the preprocessed image horizontally and find the number of ON pixels in each row.
2. Plot the histogram in y direction for the ON pixel count for the image.
3. Scan the histogram projection to find first ON pixel count with zero and remember that y coordinate as y1.
4. Continue scanning the histogram projection then we would find lots of ON pixel counts to be non zero since the characters would have started.
5. Finally we get the first ON pixel count as zero and remember that y coordinate as y2.
6. Scan the image from y1 to y2 rows for the segmented line.
7. Clear y1 and y2.
8. Repeat the above steps till the end of the histogram.

ii. *Word Segmentation*

As each word is separated with a distance, we use that concept for word segmentation. Scan the segmented line image vertically for word segmentation.
Steps for the line Segmentation is as follows:

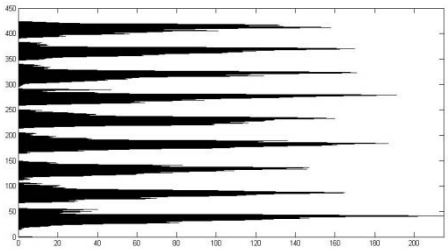
1. Scan the segmented line image vertically and find the number of ON pixels in each column.
2. Plot the histogram in x direction for the ON pixel count for the image.
3. Scan the histogram projection to find first ON pixel count with zero and remember that x coordinate as x1.

4. Continue scanning the histogram projection then we would find lots of ON pixel counts to be non zero since the characters would have started.
5. Finally we get the first ON pixel count as zero and remember that x coordinate as x2.
6. Scan the image from x1 to x2 columns and get the segment word.
7. Clear x1 and x2.
8. Repeat the above steps till the end of the vertical histogram.

iii. *Character Segmentation*

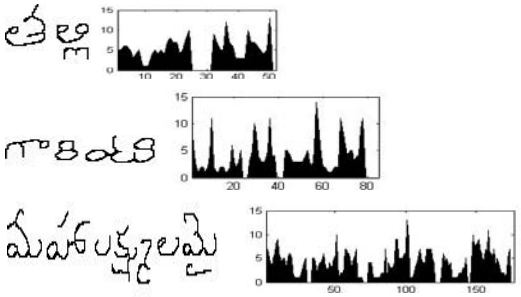
Repeat the same algorithm defined in 3.2.2 for segmenting the word into characters. In step 1 give the input the segmented word image and in step 3 use a character separating distance (as 2) based on the histogram. After completing step 8 we will be having the segmented characters.

(a) (b)

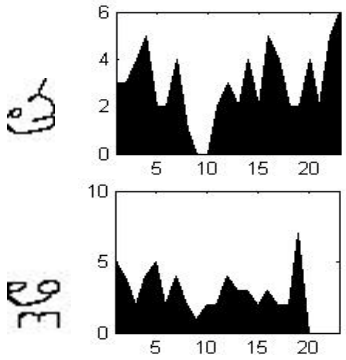


(c)

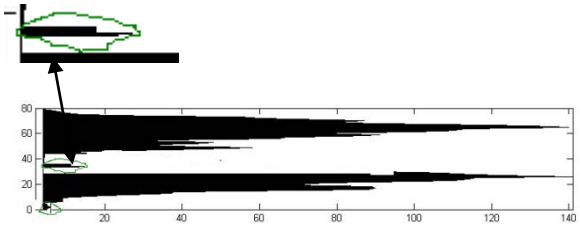
(d)



(e)



(f)



(g)

Figure 4 : Intermediate stages: (a) Input mage, (b) Pre processed step, (c) Y histogram projection (d)Text line separation with horizontal histogram projections, (e) x histogram projections for segmented words.(f) X histogram projection for segmented characters. (g)False line

c) False Line Exclusion

This procedure tries to exclude possible noises close to the text line regions. Once the possible text line are segmented by removing an offset from the histogram, we calculate the average height of these line regions to exclude false lines that might be detected. Figure 4.g a small peak in the histogram shown in green, if this region has enough height it can be confused with a text line segment by the algorithm. The equation below provides the average height of the lines found in a histogram:

$$\sum(ymax - ymin)/Nr$$

Where Ymax is the max height of the text line region and Ymin is the beginning of text region and Nr is the total no of line regions.

The lines with height below a pre-determined threshold are removed. The value of this threshold is proportional to the average height of the text lines in the whole image.

d) False Word Exclusion

As in 3.3 we will find the average height of the word in x direction and the word not satisfying the determined threshold will be treated as false word.

IV. Performance Evaluation

The performance is evaluated by checking the count of number of matches between the segmented entities with that of entities in the ground truth [16].A Match Score table is created where the pixels of the segments and the ground truth are coincide. Let *I* be the set of all image points, *Gj* the set of all points inside the *j* ground truth region, S*i* the set of all points inside the *i* segmented region, *T(s)* a function that counts the elements of set *s*. Matching results of the *j* ground truth region and the *i* segment region:

$$\text{Match Score}(i,j) = \frac{T(Gj \cap Si \cap I)}{T(Gj \cup Si \cup I)}$$

A one-to-one match is used if the matching score is equal to or above the evaluator's acceptance threshold *Ta*. If G is the count ground-truth elements, S is the count of result elements, and *o2o* is the number of one-to-one matches, we calculate the detection rate (*DR*) and recognition accuracy (*RA*) as follows:

$$DR = \frac{o2o}{G}, \quad RA = \frac{o2o}{S}$$

DR and RA is used to extract the performance metric which is

$$PM = \frac{2DR.RA}{DA+RA}$$

V. Results and Discussion

The algorithm is implemented in MATLAB. The algorithm is tested with several document images. Sample test results are shown in Figure 4.From the experiment the proposed method is fast and reliable to even for handwritten documents which have non overlapped lines. The line segmentation accuracy with DR is 99% and RA is 98% for good quality documents. The limitation of this method is that it resulted in segmentation errors for touching characters.

	M	o2o	DR(%)	RA(%)	PM(%)
Words	4044	3975	98.54	98.29	98.42
Characters	31197	27078	91.12	86.80	88.91

VI. Conclusion and Future Work

In this experiment, the proposed algorithm is tested with several document images. Even though this algorithm provides robust results it could not accurately segment the overlapped lines. A heuristic algorithm needs to be thought of in case of overlapping lines and words to recover the loss text.

References Références Referencias

1. C. V Lakshmi, C. Patvardhan. (2004): An optical character recognition system for printed Telugu text, Pattern Analysis & Applications, Volume 7, pp. 190-204.
2. Agarwal, David Doermann. (2009): Voronoi++: A Dynamic Page Segmentation approach based on Voronoi and Docstrum features, 10th International Conference, ICDAR.
3. K.S. Sesh Kumar, A. M. Namboodiri, C.V. Jawahar. (2006): Learning Segmentation of Documents with Complex Scripts, Fifth Indian Conference on Computer Vision, Graphics and Image Processing, Madurai, India, LNCS 4338, pp.749-760.
4. B.M. Sagar, DR. G. Shoba, DR. P. Ramakanth Kumar. (2008): Character Segmentation algorithms for kannada optical character Recognition, Proceedings of the 2008 International Conference on Wavelet Analysis and Pattern Recognition.
5. U. Pal and B.B. Chaudhuri, "Indian script character recognition: A Survey", Pattern Recognition, vol. 37, pp. 1887-1899, 2004.
6. B. B. Chaudhuri and U. Pal, "A complete printed Bangla OCR system", Pattern Recognition, vol.31, pp.531-549, 1998.
7. 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.
8. U. Pal and Sagarika Datta, "Segmentation of Bangla Unconstrained Handwritten Text", Proc. 7th Int. Conf. on Document Analysis and Recognition, pp.1128-1132, 2003.
9. K. Wong, R. Casey and F. Wahl "Document Analysis System ", IBM j. Res. Dev., 26(6), pp. 647-656, 1982.
10. 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.
11. 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.
12. U. Pal, B.B. Chaudhuri. (2004): Indian script character recognition: a survey, Pattern Recognition, 37, 1887 – 1899.

13. B. Anuradhaand, Arun Agarwal and C. Raghavendra Rao. (2008): An Overview of OCR Research in Indian Scripts, IJCSES, Vol.2, No.2.
14. U. Pal and B.B. Chaudhuri, "Indian script character recognition: A Survey", Pattern Recognition, vol. 37, pp. 1887-1899, 2004.
15. N. Otsu. (1979): A threshold selection method from gray-level histograms, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, VOL. SMC-9, NO.
16. I. Phillips, A. Chhabra, "Empirical Performance Evaluation of Graphics Recognition Systems", *IEEE Trans. of Patt. Analysis and Machine Intell.,* Vol. 21, No. 9, September 1999, pp. 849-870.

This page is intentionally left blank