Text - Image Separation in Devanagari Documents

Swapnil Khedekar Vemulapati Ramanaprasad Srirangaraj Setlur
Venugopal Govindaraju
Center of Excellence for Document Analysis and Recognition,
University at Buffalo, Buffalo, NY - 14260
{khedekar, raman-v, setlur, govind}@cedar.buffalo.edu

Abstract

In this paper we present a top-down, projection-profile based algorithm to separate text blocks from image blocks in a Devanagari document. We use a distinctive feature of Devanagari text, called Shirorekha (Header Line) to analyze the pattern produced by Devanagari text in the horizontal profile. The horizontal profile corresponding to a text block possesses certain regularity in frequency, orientation and shows spatial cohesion. The algorithm uses these features to identify text blocks in a document image containing both text and graphics.

1. Introduction

Given a document image, the end result of a document segmentation algorithm, in general, produces a hierarchical structure that captures the physical layout and the logical meaning of the input document image [1, 5]. The top of this structure presents an entire page and the bottom includes all glyphs on the document. The text blocks, lines, words and characters are placed at different levels in the structure.

Techniques for page segmentation and layout analysis are broadly divided in to three main categories: top-down, bottom-up and hybrid techniques [8].

Top-down techniques start by detecting the highest level of structure (large scale features like images, columns) and proceed by successive splitting until they reach the bottom layer (small scale features like individual characters). For this type of procedures, a priori knowledge about the page layout is necessary. It relies on methods such as Run-length smearing [3, 4] Projection profile methods [6, 7, 9], white streams [10], Fourier Transform [11], Template [12], Form Definition Language[13, 14], Rule-based systems [7], Gradient [15], etc.

Bottom-up methods start with the smallest elements (pixels), merging them recursively in connected components or regions (characters and words), and then in larger structures (columns). They are more flexible but may suffer from accumulation of errors. It makes use of methods like Connected Component Analysis [16, 17], Run-Length smoothing [4], Region-growing methods [18], Neighborhood-Line density [19] and Neural networks [23]. Most of these methods require high computation.

Many methods do not fit in to both of these categories and therefore are called Hybrid methods. Among these methods are Texture-based [20, 21], Gabor Filter [22]. In this paper we present a top-down technique based on horizontal profile of the image.

1.1. Some features of Devanagari script

Devanagari script has 5 basic vowels and 29 consonants along with 12 modifiers. Script has it's own specified composition rules for combining vowels, consonants and modifiers. Modifiers are attached to the top, bottom, left or right side of other characters. All characters of a word are glued together by a horizontal line, called *Shirorekha*, which runs at the top of core characters. This *Shirorekha* is the most dominating horizontal line in a text line. Top modifiers are placed above *Shirorekha* and bottom modifiers are placed below character. Figure 1 shows an example of a *Devanagari* document image.

2. Algorithm

The algorithm presented in this paper takes advantage of the *Shirorekha* which is a distinctive feature of the *Devanagari* Script. This algorithm presents a top-down approach by initially considering the complete document and trying to recursively extract text blocks from it.

2.1 Concept

The algorithm first generates the horizontal histogram of the entire image. The horizontal histogram is the number of

Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)
0-7695-1960-1/03 $17.00 © 2003 IEEE

black pixels on each row. When the horizontal histogram is plotted of the document image, the rows where *Shirorekha* is present will have maximum number of black pixels. Thus in an horizontal histogram of a *Devanagari* document, each text line will produce a pattern consisting of a peak, corresponding to *Shirorekha* and lower histogram values (number of black pixels) in it's neighborhood, corresponding to characters on lower side and ascenders on upper side of this peak. Thus the patterns (profiles) formed by text blocks are characterized by ranges of thick black peaks, separated by white gaps. Any graphics or images by contrast have relatively uniform pattern, with no prominent peaks. This method is largely independent of font size as no heuristic threshold value is used. Another main feature of a text block is that the histogram corresponding to that block possesses certain frequency and orientation and shows spatial cohesion i.e. adjacent lines are of similar height. Since text lines appear adjacent to each other, we look for adjacent patterns which have identical shape and size distribution. If this profile is found,then this portion of document must contain *Devanagari* text. Any variation from this characteristic, must be an image with or without surrounding text.

Once all the text blocks in the document are located, we need to sort the text blocks in reading order. At this stage, a prior knowledge of page layout is necessary. We present a method for a newspaper or journal article layout (from left to right and within each column from top to bottom). Simple modifications can be applied for other layouts.

2.2 Algorithm Description

- **Step 1** If the document is a grey scale image, then binarize the document. If the document is skewed, apply skew correction.
- **Step 2** Find the horizontal histogram of the document.
- **Step 3** Locate all the minima in the histogram.
- **Step 4** Cluster the histogram in to blocks so that each block is classified as either a regular block or an irregular block.

  Blocks of image whose histograms exhibit the following criteria are classified as regular blocks. All other blocks are classified as irregular blocks.

  – **Shape and Width** For every *Devanagari* text line, the portion above the *Shirorekha* contains the ascenders. Usually few characters in a line have ascenders thus the number of black pixels in these rows is less. Then there is a large increase in histogram since *Shirorekha* is the most dominant horizontal line in text. In the portion below the *Shirorekha*, characters are present. Since characters contribute fewer pixels in each row, the number of black pixels in this portion decreases drastically from *Shirorekha* portion. Thus each text line produces a shape in histogram, in which the height of graph increases drastically after some rows and then decreases to a lower value.

  Printed text consists of characters with approximately same size and thickness and are located at a regular distance from each other [2]. In a single block, usually the text is of same font size and spacing. Thus the pattern produced by each text line in horizontal histogram has almost the same width (number of rows) and is equally spaced.

  – **Intervals between peaks:** Since the *Shirorekha* is present at the top of each word, adjacent *Shirorekha*s are equally distanced in the document, due to spatial cohesion [2]. Thus the peaks in horizontal histogram corresponding to adjacent *Shirorekha*s repeat after regular intervals. This type of regularity is very rare in any image. It is a special characteristic of *Devanagari* Script.

  – **Width of the Peaks :** Since adjacent text is of same font type and size, the thickness of *Shirorekha* of each text line will be the same, producing peaks of similar widths (number of rows) in the horizontal histogram.

- **Step 5** All the regular blocks contain only text. Regular blocks are further sub divided in to text blocks based on vertical projection profile.
- **Step 6** Irregular blocks contain images. They may also contain text blocks on the left side or the right side or both sides of the image. Hence a vertical projection profile is used to sub divide these blocks in to sub blocks and the algorithm is called recursively. If the vertical histogram fails to produce any sub blocks, then that block is classified as an image block.
- **Step 7** To maintain the reading order of text in document, we need to sort the text blocks before going to next phases in recognition, so that the final output of recognition have a logical meaning [1, 5]. For this purpose, a simple approach could be to sort the text blocks according to their left boundaries as major key and top boundary as minor key.

  But sometimes this simple approach can fail if an image larger than the column size is present with text blocks on it's right. In this case, the adjacent text block might have larger left boundary as compared to other text blocks in same column, even if it's order is in between the blocks of the same column [26, 27]. To han-

dle such case, we propose to use a method on the basis of Insertion Sort using vertical overlap.

Here we maintain a queue Q of sorted blocks. Initially, Q is empty. We put first text block in Q. For every other text block T, we look for the first block in Q, say Q', which is either below or right of current block. At this point, put T before Q' in Q. The final queue will be reading order of blocks. The main complexity lies in finding the overlap between blocks to decide their positions.

Figure 1 shows an example document image with the associated histogram profile. Figure 2 shows individual text blocks obtained after applying the algorithm presented in this paper.

3. Limitations

- The irregular shaped images with non-rectangular shaped text blocks may result in loss of some text. They can be dealt with by adapting algorithms available for Roman script [28, 29].
- If there are images drawn only with sketches and not filled or very sparse images, the contribution of pixels of this image in histogram may be too less to produce irregularity in pattern produced by adjoining text. In this case, image might be misinterpreted as text.
- The simple horizontal lines in between text might be considered as text. Also the text surrounded by broad horizontal and vertical lines from all four sides, may result in detection of a complete document as single image. This problem can be solved by applying line detection algorithms [30] at the preprocessing stage

4. Conclusions

In this paper we presented an efficient algorithm for segmenting and classifying printed *Devanagari* newspaper documents. We tested this algorithm on variety of documents from different newspapers, journals, books with different page layouts. The result shows that it works satisfactorily. We also presented a sample newspaper image and it's result.

References

[1] J. Liang, I.T. Phillips, R.M. Haralick *A Statically Based, Highly Accurate Text-line Segmentation Method.* Proc. 5th Intl. Conf. on Document Analysis and Recognition. ICDAR'99, Bangalore, India, pp. 551.

[2] Q. Yuan, C.L. Tan *Text Extraction from Gray Scale Document Images Using Edge Information.* Proceedings of the International Conference on Document Analysis and Recognition, ICDAR'01, September 10-13, 2001, Seattle, USA, pp. 302-306.

[3] J. Kanai, M.S. Krishnamoorthy, T. Spencer *Algorithm for Manipulating nested block represented images.* SPSE's 26th Fall Symposium, Arlington VA, USA, Oct 1986, pp.190-193.

[4] K.Y. Wang, R.G. Casey, F.M. Fahl *Document analysis system.* IBM Journal of Research and Development, Vol. 26, No. 6, Nov 1982, pp. 647-656.

[5] G. Nagy, S. Seth*Hierarchical Representation of Optically Scanned Documents.* Proc. of 7th Intl. Conf. on Pattern Recognition,Montreal, Canada, 1984, pp. 347-349.

[6] M.S. Krishnamoorthy, G.Nagy, S. Seth, M. Vishwanathan *Syntactic Segmentation and Labeling of Digitized Pages from Technical Journals.* IEEE Computer Vision, Graphics and Image Processing, Vol. 47, 1993, pp. 327-352.

[7] K.H. Lee, Y.C. Choy, S. Cho *Geometric Structure Analysis of Document Images: A Knowledge-Based Approach.* IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 22, Nov 2000, pp. 1224-1240.

[8] O. Okun, D. Doermann, Matti P. *Page Segmentation and zone classification.* The State of the Art, Nov 1999.

[9] O. Iwaki, H. Kida, H. Arakawa *A Segmentation Method based on Office Document Hierarchical Structure.* Proc. of Intl. Conf. on Systems, Man and Cybernetics, Alexandria VA, USA, Oct 1987, pp. 759-763.

[10] T. Pavlidis, J. Zhou *Segmentation by White Streams.* Proc. Intl. Conf. on Document Analysis and Recongition, ICDAR'91, St-Malo, France, pp. 945-953.

[11] M. Hose and Y. Hoshino. *Segmentation method of document images by two-dimensional Fourier transformation.* System and Computers in Japan, Vol. 16, No. 3, 1985, pp. 38-47.

[12] A. Dengel and G. Barth. *Document description and analysis by cuts.* Proc. Conference on Computer-Assisted Information Retrieval , MIT USA, 1988.

[13] H. Fujisawa and Y. Nakano. *A top-down approach for the analysis of document images.* Proc. of Workshop on Syntactic and Structural Pattern Recognition (SSPR' 90), 1990, pp. 113-122.

[14] 11. J. Higashino, H. Fujisawa, Y. Nakano and M. Ejiri. *A knowledge-based segmentation method for document understanding.* Proc. 8th Intl. Conf. on Pattern Recognition, Paris, France, 1986, pp. 745-748.

[15] J. Duong, M. Ct, H. Emptoz, C. Suen. *Extraction of Text Areas in Printed Document Images.* ACM Symposium on Document Engineering ,DocEng'01, Atlanta (USA), November9-10, 2001, pp. 157-165.

[16] J. P. Bixler. *Tracking text in mixed-mode document.* Proc. ACM Conference on Document Processing System, 1998, pp. 177-185.

[17] H. Makino. *Representation and segmentation of document images.* Proc. of IEEE Computer Society Conference on Pattern Recognition and Image Processing, 1983, pp. 291-296.

[18] A.K. Jain, Fundamentals of Digital Image Processing. Prentice Hall USA, 1989.

[19] O. Iwaki, H. Kida and H. Arakawa. *A character / graphics segmentation method using neighborhood line density.* Trans. of Institute of Electronics and Communication Engineers of Japan, 1985, Part D J68D, 4, pp. 821 828.

[20] D. Chetverikov, J. Liang, J. Komuves, R.M. Haralick. *Zone classification using texture features.* Proc. of Intl. Conf. on Pattern Recognition, Vol. 3, 1996, pp. 676-680.

[21] W.S. Baird, S. E. Jones, S. J. Fortune. *Image segmentation by shape directed covers.* Proc. of Intl. Conf. on Pattern Recognition, Vol. 4, 1996 pp. 820-825.

[22] A. K. Jain and S. Bhattacharjee, *Text Segmentation Using Gabor Filters for Automatic Document Processing,* Machine Vision and Applications, Vol. 5, No. 3, 1992, pp. 169-184.

[23] C.L. Tan, Z. Zhang *Text block segmentation using pyramid structure.* SPIE Document Recognition and Retrieval, Vol. 8, January 24-25, 2001, San Jose, USA, pp. 297-306.

[24] C. Strouthopoulos, N. Papamarkos. *Text identification for document image analysis using a neural network* Image and Vision Computing, 1998, Vol. 16, Iss. 12-13, pp. 879-896.

[25] R.M.K. Sinha, V. Bansal, *A Devanagari OCR and A Brief Overview of OCR Research for Indian Scripts* Proceedings of STRANS'01, IIT Kanpur, India, 2001.

[26] D. Wang, S. Srihari *Classification of Newspaper Image Blocks Using Texture Analysis.* Computer Vision , Graphics, and Image Processing, Vol. 47, 1989, pp.327-352.

[27] D. Niyogi, S.N. Srihari *Knowledge-Based Derivation of Document Logical Structure.*

[28] V. Wu, R. Manmatha, E. M. Riseman *Finding Text in Images.* Proc. of 20th International ACM SIGIR Conference on Research and Development in Information Retrieval, Philadelphia, 1997, pp. 3-12.

[29] M.Y. Hasan and L.J. Karam, *Morphological Text Extraction From Images.* IEEE Trans. on Image Processing, vol. 9, Nov. 2000, pp. 1978-1983.

[30] P. Mitchell, H. Yan *Newspaper document analysis featuring connected line segmentation* Proc. Intl. Conf. on Document Analysis and Recognition, ICDAR'01, Seattle, USA.

Figure 1. Sample Image.

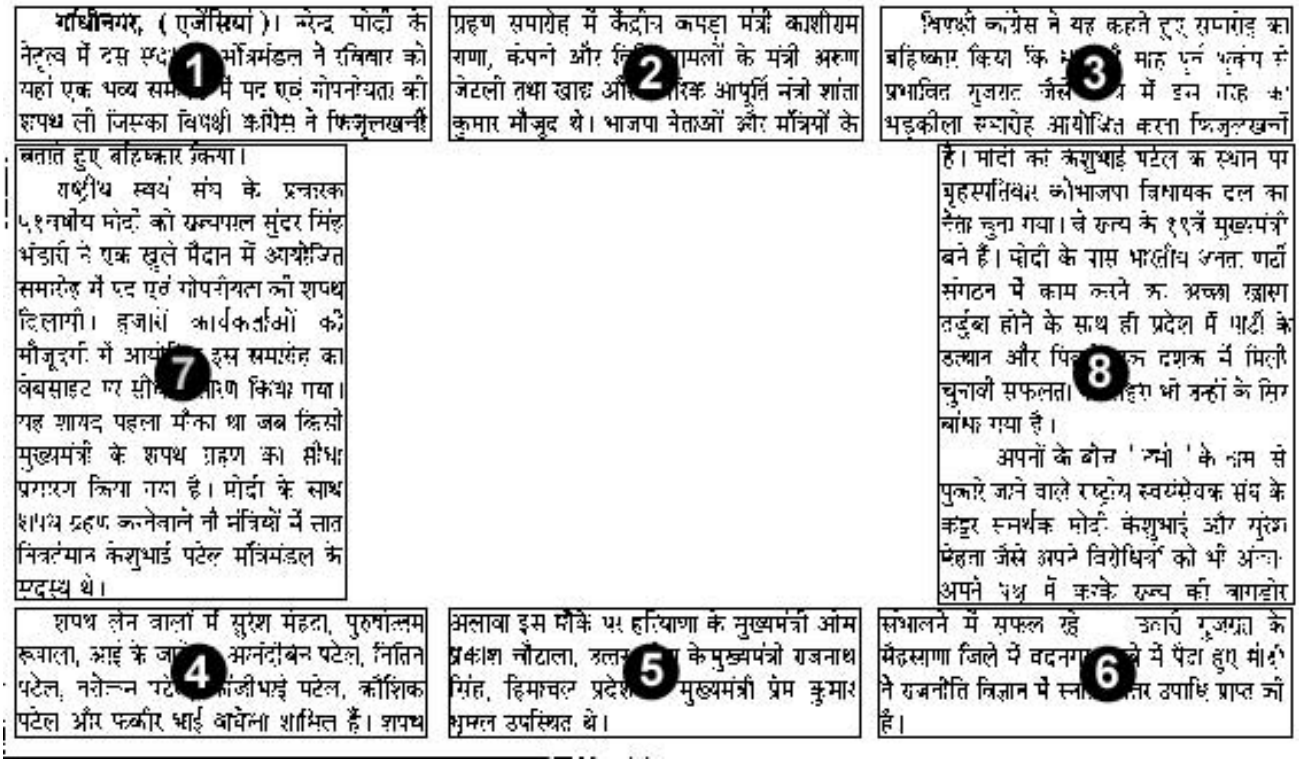

Figure 2. Resulting text blocks before re-ordering.