DEVANAGIRI DOCUMENT SEGMENTATION USING HISTOGRAM BASED APPROACH

Neha Sahu
Amity School of Engineering & Technology, Noida

Abstract

Text in images is a significant cue for visual content, understanding and retrieval. Detection and extraction of text in images have been used in many applications. Document segmentation is one of the critical phases in machine recognition of any language. Correct segmentation of individual symbols decides the accuracy of character recognition technique. It is used to decompose image of a sequence of characters into sub images of individual symbols by segmenting lines and words. Devnagari is the most popular script in India. It is used for writing Hindi, Marathi, Sanskrit and Nepali languages. Moreover, Devnagari documents consist of vowels, consonants and various modifiers. Hence proper segmentation of Devnagari word is challenging. A simple histogram based approach to segment Devnagari documents is proposed in this paper. Various challenges in segmentation of Devnagari script are also discussed.

Keywords : Line segmentation, Devanagiri, Character segmentation, word Segmentation

1. Introduction

Machine simulation of human functions has been a challenging research field since the advent of digital computers. In some areas which require certain amount of intelligence such as number crunching or chess playing, tremendous improvements are achieved. On the other hand, humans still outperform even the most powerful computers in the relatively routine functions such as vision. Imitating the human abilities in computers is not an easy task due to high context sensitivity[2][3]. Machine simulation of human reading is one of these areas which have been the field of intensive research from early days of computer. Yet it is still far from the final frontier. Optical Character recognition (OCR) and document processing has become the need of time with the popularization of desk top publishing and usage of internet. OCR also finds application in postal mail sorting, Automatic Bank cheque clearance, Blind person aid, automatic form processing. OCR involves recognition of characters from digitized images of optically scanned document pages. The characters thus recognized from document pages are coded with American Standard Code for Information Interchange (ASCII) or some other standard codes like UNICODE for storing in a file which can be further edited as any other file created with some word processing software or editor. In Optical Character Recognition (OCR), the text lines, words and symbols in a document must be segmented properly before recognition. Correctness/ incorrectness of text line segmentation directly affect accuracy of word/character segmentation and consequently affect the accuracy of word/character recognition. Several techniques for text line segmentation are reported in the literature[4]. These techniques may be classified 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.
Piece-wise horizontal projection analysis of black pixels is employed by many researchers to segment text pages of different languages[1][5]. In piecewise horizontal projection technique, the text-page image is decomposed into horizontal stripes. The positions of potential piece-wise separating lines are obtained for each stripe using 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 such as skew detection, slant detection, text line segmentation, etc. Thinning operation is also used by researchers for text line segmentation from documents[8].

2. Characteristics of Devanagari Script

Devanagari is most popular script to write Hindi as well as Sanskrit, Marathi, Sindhi, and Nepali language with minor modifications. The alphabets of Devanagari script consists of 33 consonants and 14 vowels. There is no concept of lower or upper case in Hindi language. In Devanagari script, a text word may be partitioned into three zones. The upper zone denotes the portion above the headline, the middle zone covers the portion of basic and compound characters below the headline, and the lower zone may contain where some vowel and consonant modifiers can reside. For a long number of characters (basic as well as compound) there exists a horizontal line at the upper part called “shirorekha” or headline in Hindi. The imaginary line separating the middle and lower zone may be called the base line. In Hindi language characters also have a half form which increases the language complexity for recognition. The half characters may touch with full characters to make the characters called conjuncts[10][11]. Two consecutive lines touches or overlap each other due to these modifiers. This makes the segmentation of handwritten Hindi text very complex. Devnagari has 11 vowels and 33 consonants. They are called basic characters. Vowels can be written as independent letters, or by using a variety of diacritical marks which are written above, below, before or after the consonant they belong to. When vowels are written in this way they are known as modifiers and the characters so formed are called conjuncts. Sometimes two or more consonants can combine and take new shapes. These new shaped clusters are known as compound characters. These types of basic characters, compound characters and modifiers are present not only in Devnagari but also in other scripts[1][4]. All the characters have a horizontal line at the upper part, known as Shirorekha. In continuous handwriting, from left to right direction, the shirorekha of one character joins with the shirorekha of the previous or next character of the same word[5][6]. In this fashion, multiple characters and modified shapes in a word appear as a single connected component joined through the common shirorekha. Also in Devnagari there are vowels, consonants, vowel modifiers and compound characters, numerals. Moreover, there are many similar shaped characters. All these variations make Devnagari Optical Character Recognition, a challenging problem.

A simple Devanagiri script is shown in the figure below[7].

![Figure 2.1 Vowels of the contemporary Devanagari script](image-url)
3. Segmentation

Segmentation is one of the most important phases in character recognition process. Segmentation is the process of segmenting the whole document image into recognizable units. The segmentation process is divided into four major parts.

i. Line segmentation  
ii. Word segmentation  
iii. Zone segmentation  
iv. Character segmentation

3.1 Line Segmentation

The first step of segmentation process is segmenting the text region into lines, also called as line segmentation. Line segmentation as shown in the figure below. Header lines are rows with maximum number of black pixels and base lines are rows with minimum number of black pixels. Finding header line is a challenge because of skew in headline[8]. Till now most of the researchers are detecting the header line by finding the row with maximum pixel density, but it cannot work for skew variable text.

The algorithm proposed by M. K. Jindal helps to find header line and base line [16] [17]. This method gives good results for uniform and nonuniform skewed lines. But average line height is assumed to be 30 pixels. As different users handwritings are different, so average line height of users may be greater than 30 pixel or less than 30 pixels. Average line height of high resolution images is greater than low resolution images. If 450x540 image has average line height as 30 pixels, 900x1080 image average line height will be 60 pixels, 225x270 image average line height will be 15 pixels. In the above two situations this algorithm fails to detect the lines accurately. The new algorithm can work for the images having different average line height. In this algorithm we need to find average line height using horizontal projection based method. Divide the Image into three equal halves (stripes), because the skew in entire line may be high, as compared to skew in the first half. Perform the following steps in first half[10][12][13].

1) Find out the rows with minimum number of pixels and replace that row pixels with black pixels. So that we can separate text lines with black rows.  
2) Find out the height of the text lines using those black rows.  
3) Store the heights of lines in array and use sorting technique to sort the elements and take median as average height of the lines.  
4) By using average line height calculate minimum height of the consonant in a line. Minimum height of the consonant = Average line height / 4.

3.2 Word Segmentation

Word segmentation is easier than line segmentation and character segmentation. Space between two words is generally more than three pixels. Words are segmented by the projection based method[4][5].
- Construct the Vertical Histogram for the image
- Count the white pixel in each column.
- Using the Histogram, find the columns containing no white pixel.
- Replace all such columns by 1
- Invert the image to make empty rows as 0 and text words will have original pixels.
- Mark the Bounding Box for word.
- Copy the pixels in the Bounding Box and save in separate file.

3.3 Character Segmentation
- Get the thinned image using Matlab bwmorph function. (This is done to normalize image against thickness of the character).
- Count the white pixel in each column.
- Find the position containing single white pixel.
- Replace all such columns by 1.
- Invert the image to make such columns as 0 and text characters will have original pixels.
- Mark the Bounding Box for characters using standard Matlab functions.
- Copy the pixels in the Bounding Box and save in separate file.

4. Results and Discussion

Figure 4.1 Original Text Image

Figure 4.2 Line Segmentation

Figure 4.3 Line Segmentation

Figure 4.2 Word Segmentation

Figure 4.2 Word Segmentation

Figure 4.2 Word Segmentation

Figure 4.2 Character Segmentation
Table 4.1 Overall Segmentation results for document

<table>
<thead>
<tr>
<th>Line Segmentation</th>
<th>Lines in Document</th>
<th>Recognized lines</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4</td>
<td>4</td>
<td>10%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Word Segmentation</th>
<th>Words in document</th>
<th>Recognized words</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20</td>
<td>18</td>
<td>90%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Character segmentation</th>
<th>Characters in document</th>
<th>Recognized characters</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>77</td>
<td>60</td>
<td>77%</td>
</tr>
</tbody>
</table>

References


