Novel Approach for Segmenting Fused / Merged Characters During Character Segmentation

ABDUL MOMIN
dudemomin12@yahoo.co.in
Indian Institute of Information Technology Allahabad
Deoghat, Jhalwa, Allahabad, Uttar Pradesh – 211012
INDIA

SHARAD GUPTA
sharad311285@gmail.com
Indian Institute of Information Technology Allahabad
Deoghat, Jhalwa, Allahabad, Uttar Pradesh – 211012
INDIA

SUDIP SANYAL
ssanyal@iiita.ac.in
Indian Institute of Information Technology Allahabad
Deoghat, Jhalwa, Allahabad, Uttar Pradesh – 211012
INDIA

Abstract: - This paper examines various approaches for handling the problem of overlapped or fused / merged characters while performing character segmentation of scanned document images. A three pass method is proposed that involves a naïve profile based segmentation followed by methods to resolve overlapped characters and fused / merged characters. It is found that erosion of white regions gives excellent results for separating fused / merged characters. Experiments performed on a large number of pages clearly demonstrate the efficacy of the proposed method.

Key-Words: - Optical character recognition, character segmentation, overlapped characters, fused or merged characters.

1 Introduction
As is well known, conversion of a scanned image of text into an editable form requires an Optical Character Recognition (OCR) system. One of the key steps in an OCR system is the process of segmenting the scanned image of the document into images of the individual characters. The segmented character images are then passed through a pattern recognition system that has been trained to recognize the characters of a particular script.

There is a large body of literature on the pattern recognition step. The focus of the current work is on the segmentation step. In particular, we focus on the alleviation of problems faced when the original document is old or the scan quality is poor. This problem is of practical interest because we usually do not need to OCR new documents since they are “born digital” i.e. they have been produced using word processors. Thus, OCRs are primarily useful when we are dealing with old documents.

Techniques for character segmentation depend on the fact that there is a white space around every character. Thus, one can broadly categorize the character segmentation process into three techniques [1]. Most of the approaches also assume that the image has been binarized and contains a black foreground and white background.

a) Bottom-up techniques like object labelling. This method looks for the connected components in the images and labels the blobs [2] [3].

b) Top-down techniques like horizontal and vertical profiling. This method finds the number of black pixels in each row (horizontal profiling). The lines are segmented by looking for zeros in the
horizontal profile. For each line the method then finds the number of black pixels in each column (vertical profiling). Words and characters are segmented by looking for zeros in the vertical profile.

c) Holistic approach.

Given clean inputs (which are typically scans of new documents) these approaches work quite well since the characters are well separated. However, for old documents, this is frequently not the case. Due to the porosity of the paper, the ink tends to spread with time. Moreover, with aging, the paper becomes yellowish leading to a non-uniform background. These phenomena lead to a situation where two neighbouring characters may touch at some point. The character segmentation process fails at these points since there is no white space around the individual characters. This leads to fused or merged characters. Examples of such merged characters are shown in Figure 1 below for some old Tibetan documents. It is obvious that neither of the two characters will be recognized properly by the subsequent pattern recognition module. We have observed in our experiments that books that have been printed around 60 – 70 years ago can have ~ 25% characters forming merged or fused pairs. Thus, this problem can contribute significantly in degrading the performance of an OCR.

Figure 1:- Examples of fused or merged characters

In the present work we have proposed an approach to handle the problem of separating fused or merged characters. We have performed our experiments on printed Tibetan script even though we expect that similar approach will work for other scripts too.

A secondary problem that we have observed in Tibetan and other Indic scripts, is that of overlapped characters. We frequently observe that the upper (lower) vowel modifier of a character protrudes over (under) the neighbouring character. Thus, profile based segmentation methods will not be able to find a zero in the vertical profile. This will result in the two characters to behave like fused or merged characters even though they may not touch at any point. Examples of such overlapped characters are shown in Figure 2 below. The approach proposed in the present work can solve this problem too.

Figure 2:- Examples of overlapped characters

In the next section we have described some earlier efforts to solve the abovementioned problem. Our approach has been presented in section 3. Section 4 describes our experiments and the results thereof. The last section concludes the paper.

2 Previous work

While most OCRs would have a module that addresses the above issues to some extent, there are very few reports that focus exclusively on the segmentation issues. Bayer et al [4] proposed a vertical cut approach to segment fused characters. This is a very efficient approach when the characters are not overlapping. But most unfortunately in Tibetan and other Indic scripts, the overlapped characters occur frequently. Das [5] have proposed a method for handling overlapped characters but have not considered the problem of fused characters. Bansal and Sinha [6] worked on touching and fused characters in Devnagari script and proposed a two stage algorithm. They maintain a statistical record of the height and width of the characters. In the first level they segment each line and character and then identified the composite and fused characters using the statistical record. But in our context the character height and width are varying even in a single page for the same character. Thus, the above approach is not suitable for such scripts. Fujisawa et al [7] use a recognition based approach. Kumar et al [10] discuss how to extract a character from a line and then they have tried to see if the extracted character is meaning full. Furferi et al [11] have proposed a pixel cloud method. This can be introduced to form a model for each character to segment fused and merged character’s problem. We are proposing an approach where we will run the segmentation process in three levels or passes to deal with overlapped characters and fused / merged characters.

3 Approach
We present a method where we will consider both fused / merged characters and overlapped characters. While our testing has been performed primarily on printed Tibetan scripts, we expect that the same process will work well on other scripts. We assume that the scanned images are in binary form and skew free. The testing has been performed on documents scanned at a resolution of 600 dpi.

As mentioned earlier, our approach uses three passes. In the first pass we segment the characters using profiling. We use connected component analysis in the second pass to segment the overlapped characters and in the third pass we deal with merged characters using dilation. Thus, in the first pass we perform a crude segmentation of the characters. This segmentation will contain merged and overlapped characters. The second and third passes attempt to correct the errors of the first pass.

Thus, after the first pass our first task is to detect the instances of overlapped or merged characters cases. Once such cases have been reliably identified we apply corrective measures to deal with these cases.

At this stage it is necessary to point out a characteristic of Tibetan and several other Indic scripts like Devanagari, Bangla etc. Each symbol has a horizontal line called the shiro-rekha. The main character is below the shiro-rekha while there may be some vowel modifiers above the shiro-rekha as shown in Figure 3 below. The shiro-rekha needs to handled carefully during segmentation. In the following we describe each of the passes in detail.

3.1 Pass 1:
In the first pass we have segmented lines from the image using horizontal profiling (Figure 4) and then using vertical profiling on each line we segment the upper modifiers (above_shiro) and characters (below_shiro). After this we segment characters (below_shiro) using horizontal profiling (Figure 5).

![Figure 3: Tibetan script (shiro_rekha, above_shiro, below_shiro)](image)

3.2 Pass 2:
In our test pages the character widths are not fixed. Moreover, the same character may have significant variability in widths in a single page. So we have created a table where we collect the average width of each character’s possible widths, height and aspect ratio using Tibetan Data Set created by us. The threshold of permissible widths is the average of the actual widths (excluding the punctuation marks).

After the naïve character segmentation as described in pass 1, we have a run a width checker to check the width of each and every segmented object. If the tested object width is higher than the threshold then this is a suspected overlapped or merged character case.

Let us examine an example of overlapped character and merged characters.

![Figure 6:- (a) Overlapped characters (b) merged characters](image)
Now after identifying the suspected overlapped or merged character case we run a connected component analysis [2][3] on the suspect list. The connected component analysis will reveal that the example (a) above is made of two components. Moreover, both the components now have widths that are within the acceptable range. Thus, they will be removed from the suspect list and added to the list of well segmented characters. On the other hand, in the case of (b) above the object will stay in the suspect list and are candidates for the third pass.

![Figure 7:- (a) Original overlapped characters (b) character segmented after connected component analysis.](image)

3.3 Pass 3:
In order to deal with merged characters we have tried several methods. The first approach was to consider the merge point as the one corresponding to the minima in the vertical profile. However, this approach did not give correct estimates of the merge point in many cases and thus was rejected.

The second approach was to perform morphological operations like dilation or erosion [9]. Within this category we had two options – either erode the black region or dilate the white region. It is important to point out that these are not equivalent operations.

If A is the binary image and B is the structuring element which is a 3x3 square matrix, then the elements of B will be $B = \{(-1,-1), (-1,0), (-1,1), (0,-1), (0,0), (0,1), (1,-1), (1,0), (1,1)\}$.

Erode: The erosion is a skeletal operator that deflates a binary object.

$$A \ominus B = \bigcap_{b \in B} A_b$$

Dilate: The dilation is a stretching operator that enlarges a binary object.

After analysing the effect of these two operations on a large number of samples we find that dilation of white region is more effective than the erosion of black region. The sample image (Figure 7a) contains two merged characters. Figure 7b shows the result of erosion of black pixels while 7c gives the results of dilation of the white region. As can be seen, the merging of characters is still visible in 7b while the characters have separated in 7c.

![Figure 8:- (a) Original image of merged character case (b) after erosion (c) after dilation](image)

4 Results and discussion
The approach described above has been tested for a large number of pages. Below we present the results for four sample pages. After each pass we have run a manual analysis to check what the output of the each task is in order to have a numerical measure of accuracy. The results are summarized in Table 1 below. The first column gives the actual number of characters that the page contains. The second column shows the number of characters obtained after pass 1 (i.e. the initial profile based segmentation). The third shows the improvements when the overlapped characters are removed using connected component analysis. The fourth column gives the number of characters obtained by trying to solve the merged character problem using the erosion (of black region) approach. The last column depicts our final result when we try to solve the problem of merged characters using the dilation (of white region) approach.
From Table 1 it is clear that the number of characters obtained after the first pass is less than the original because the merged character and overlapped characters remain. But after second pass the overlapped characters are segmented. So the total number of segmented characters increases. A thorough analysis of the fourth and fifth columns reveal that the dilation based method gives considerably better results than the erosion based method. In most of the cases we are able to recover almost all the original characters. However, in some cases it may lead to some over segmentation as in the third sample page.

5 Conclusion

In this work we have analyzed different approaches for character segmentation which is an essential step for OCRs. We have examined the case of old printed books that frequently have degraded pages. The usual segmentation algorithms lead to a large number of overlapped characters or fused / merged characters that can not be recognized by the subsequent recognition engines. Different techniques were explored to solve this problem like and dilation of the white regions gave the best results. One problem that has been observed with the above process is that the morphological operations lead to some distortion of the characters. While the distortions are not very severe, they may affect the accuracy of the recognition engine. However, the effect of the proposed approach improved the accuracy of the overall process in all the cases. Sometimes the accuracy increased by over 10% which is quite significant.

Acknowledgements: The authors would like to acknowledge the financial assistance from TDIL, MCIT and the infrastructure provided by IIIT Allahabad.

References: