Character Segmentation Using Visual Inter-word Constraints In Text Page

Tao Hong
Center for Document Analysis and Recognition
Department of Computer Science
State University of New York at Buffalo
Buffalo, New York 14260
Voice: (716) 645-3407
Fax: (716) 645-6176
Email: taohong@cedar.buffalo.edu

Jonathan J. Hull
Center for Document Analysis and Recognition
UB Commons
520 Lee Entrance, Suite 202
State University of New York at Buffalo
Buffalo, New York 14228-2567
Voice: (716) 645-6164 x109
Fax: (716) 645-6176
Email: hull@cedar.buffalo.edu

submitted to:

IS&T/SPIE’s Symposium on Electronic Imaging: Science & Technology
Conference on Document Recognition Technologies
Conference Chairs: Drs. Luc Vincent and Henry Baird

SPIE Conf. on Document Recognition II, San Jose, CA, Feb. 6-7, 1995, pp. 76-83.
ABSTRACT

For degraded text recognition by OCR, character segmentation is a critical but difficult preprocessing step. If a text page is highly degraded, it is hard to locate each segmentation point correctly because there may be a lot of touching or broken characters in the text image. Most current methods towards the problem are based on connected component analysis, aspect ratio estimation or profile analysis. Other current methods take the approach of using character recognition results and contextual information to help segmentation. In this paper, we present a method to utilize visual inter-word constraints available in the text image to split word images into smaller image pieces, which can be split further by using the same method. Visual inter-word constraints to be considered here are simply the information about whether a word image is a sub-image of another word image. For example, given two word images $A$ and $B$, which are "mathematical" and "the" respectively, we can find out that the short word image $B$ is a sub-image of the long word image $A$ by image matching, and therefore, the longer image $A$ can be split into three pieces, $A_1$, $A_2$, and $A_3$, where $A_2$ is matching to $B$, $A_1$ corresponds to "ma", and $A_3$ corresponds to "matical". And image piece $A_1$ can be further used to split $A_3$ into two parts, "ma" and "tical". After applying the method on all word images from a degrade text page, most word images will finally be segmented to character level. Although the method is purely image processing and with no recognition step involved, contextual information available in the text image is utilized.

Key Words: Character segmentation, degraded text recognition, contextual analysis, visual inter-word constraints, OCR.

1 Introduction

As a critical preprocessing step for text recognition by OCR, character segmentation is to partition word images into sequences of character images so that OCR technique can be applied later to recognize those isolated characters. The performance of a text recognition system can be heavily depend on the performance of its character segmentation step. It was reported that the majority errors in machine-printed text recognition are caused by incorrect character segmentation.

The simplest way to do character segmentation is to use the small space between characters as segmentation point. This strategy can not work well when there are touching or broken characters, which always occur in degraded text images, such as multiple-generation photocopies or facsimiles. In the situation, two or more characters may be segmented as one character image or one character image may be split into two pieces. To treat those mis-segmented image pieces as character images and to apply OCR on them, a text recognition system will generate incorrect character recognition results which may make the text recognition result unreadable at word or higher level.

To make character segmentation more accurate, many methods was developed. Many current methods towards the problem are based on connected component analysis, aspect ratio
estimation or profile analysis([1, 3, 4, 6]). There are also some methods which integrates character segmentation with character recognition under the belief that segmentation decisions are tentative until confirmed by successful recognition of the segmented image pieces and further verified by using contextual information([2, 8]).

In the paper, we present a method for character segmentation to utilize visual inter-word constraints available in the text image to split word images into smaller image pieces, which can be split further by using the same method.

Case 1:

\[
\text{there} \quad \rightarrow \quad \text{the} + \text{re}
\]

Case 2:

\[
\text{hypothesis} \quad \rightarrow \quad \text{hypo} + \text{the} + \text{sis}
\]

Case 3:

\[
\text{share} \quad \rightarrow \quad \text{sh} + \text{are}
\]

The intuition of the method came from the observation that, inside an English text, there

Figure 1: Part of text page with three sentences

Figure 2: Split a Word Image by Another Word Image

SPIE Conf. on Document Recognition II, San Jose, CA, Feb. 6-7, 1995, pp. 76-83.
are many similar patterns at image level. For example, in Figure 1, word image 2 can match very well to the right part of word image 11; word image 5 can match to the left part of word image 7, and it can also match to the middle part of word image 6; and so on. If image A can match with a part of image B, image B can be split into two or three image pieces (see Figure 2 for the examples for different cases). Those new image pieces can be used to split other larger images and themselves can be further split by other smaller images (see Figure 3 for an example which shows how the character image “a” can be obtained by splitting repeatedly. With many word images from text image at beginning, they will be partitioned into smaller and smaller pieces. Finally, most word images can be segmented into sequences of character images.

![Figure 3: A Character Image Be Segmented After Two Steps](image)

In next section, we will present the algorithm of character segmentation for the word images from a text page.

2 Algorithm Description

Visual similarity between two binary images of same size can be measured by the percentage of pixels which mismatch. Let A and B be two $m \times n$ binary images, the visual similarity between A and B can be measured by

$$r = \frac{\sum_{i}^{m} \sum_{j}^{n} (A_{ij} \oplus B_{ij})}{m \times n}$$

where “$\oplus$” is exclusive or operator. The lower the percentage $r$ is, the better two images match. It was reported that the word images from a degraded text page could be clustered correctly by using exclusive or operator([5]). The similarity between two images, A and B, with small difference in size, can be defined by the maximal matching we can get if we shift A cross B.

To check whether an image pattern A is a sub-image of another larger image B, A can be shifted on B to find maximal matching. If the $r$ score of the maximal matching is lower than a threshold, we can say A is a sub-image of B.

The segmentation algorithm is described in Figure 4. Given a text page, after layout analysis, images of line can be extracted. Because of relative large gap between word images in a line, word images can be extracted easily. The algorithm not only segments large images into smaller pieces, but also clusters the similar images into clusters. The images in a cluster are the images with same interpretation. For each image cluster, a prototype can be derived based
extract all word images from the text image;
set QUEUE as an empty set
put all word images into QUEUE
set IMAGE-CLUSTER-LIST as an empty set
while QUEUE not empty do
    pop an image I from QUEUE
    if image I matches with the prototype of a cluster in IMAGE-CLUSTER-LIST then
        add image I as a new member of that image cluster
    else if
        for each cluster in IMAGE-CLUSTER-LIST do
            if image I is a sub-image of the prototype of X
                then
                    segment images in X
                    put the new image pieces into QUEUE;
                else if
                    if the prototype of X is a sub-image of image I
                        segment image I
                        put the new image pieces into QUEUE;
                end if
        end for
        create a new image cluster C with image I
        add C into IMAGE-CLUSTER-LIST
    end if
end while

Figure 4: Algorithm of Character Segmentation by Using Inter-word Visual Constraints

on the images in the cluster. In the beginning, all word images will be inserted into a queue and there is not any image cluster. Each time, an image will be popped out of the queue to see whether it can be split or it can be used to split other images. If the image can match to the prototype of an existing cluster, it will be added into that cluster. Otherwise, a new cluster will be created for the image and the image will be used to do image partition. If the image is a sub-image of the prototype of a cluster, the prototype will be split into two or three parts and those parts will be inserted into the queue. If the prototype of a cluster is a sub-image of the image, the image will be split into two or three parts and those parts will be inserted into the queue. For each new image part, we record where it comes from. The process of clustering and segmenting will end when the queue becomes empty. Then, image clusters will be checked to maximize the number of parts that a word image can be split. In a cluster, if an image can be split in one way, the other images should also be split in the same way.
Figure 5 is a simple example to show the initial state and the final state when running the algorithm on three word images.

**Initial State**

<table>
<thead>
<tr>
<th>Queue</th>
<th>Image Cluster List</th>
</tr>
</thead>
<tbody>
<tr>
<td>mathematical</td>
<td>cluster 1:</td>
</tr>
<tr>
<td></td>
<td>cluster 2:</td>
</tr>
<tr>
<td>the</td>
<td>cluster 3:</td>
</tr>
<tr>
<td>the</td>
<td>cluster 4:</td>
</tr>
<tr>
<td>the</td>
<td>cluster 5:</td>
</tr>
<tr>
<td>a</td>
<td>cluster 6:</td>
</tr>
<tr>
<td>a</td>
<td>cluster 7:</td>
</tr>
<tr>
<td>a</td>
<td>cluster 8:</td>
</tr>
<tr>
<td></td>
<td>empty</td>
</tr>
</tbody>
</table>

**Final State**

<table>
<thead>
<tr>
<th>Queue</th>
<th>Image Cluster List</th>
</tr>
</thead>
<tbody>
<tr>
<td>empty</td>
<td>cluster 0:</td>
</tr>
<tr>
<td></td>
<td>cluster 1:</td>
</tr>
<tr>
<td>a a a a</td>
<td>cluster 2:</td>
</tr>
<tr>
<td>m m</td>
<td>cluster 3:</td>
</tr>
<tr>
<td>tic</td>
<td>cluster 4:</td>
</tr>
<tr>
<td>the the</td>
<td>cluster 5:</td>
</tr>
<tr>
<td>ma</td>
<td>cluster 6:</td>
</tr>
<tr>
<td>them</td>
<td>cluster 7:</td>
</tr>
<tr>
<td>matical</td>
<td>cluster 8:</td>
</tr>
<tr>
<td>mathematical</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5: An Example of Running the Algorithm
3 Preliminary Experimental Results And Discussion

Using an image of paragraph cut from a scanned text page (see the part (a) of Figure 6), we tested the algorithm to see how it works. The part (b) of Figure 6 is the segmentation result generated by the algorithm. In the figure, segmentation points located by the algorithm are marked by the vertical short lines. Here, the threshold of \( r \) for image matching was set to 0.15. There are 110 word images in the paragraph. There are totally more than 700 image clusters generated by the algorithm. Although there are space between most characters, we did not use this information at all to segment the characters. The word “4,000” was not segmented because the digits in the word did not appear elsewhere in the paragraph.

Contemporary linguists have argued that the ability to learn language is more than an ordinary human skill; it is biologically based. Language is something we are born knowing how to know. Yet the hypothesis that there are biological underpinnings to human linguistic ability does not explain everything. There may indeed be an innate language capacity, a so-called universal grammar, but despite the proponents of Esperanto, there is no universal language. Depending on the accidents of birth, a child may end up a native speaker of any one of roughly 4,000 languages. Thus the predisposition to acquire language seems to be remarkably flexible as well as strong.

(a)

(b)

Figure 6: A Text Block and Result of Character Segmentation

4 Conclusions and Future Directions

We presented a method of character segmentation for the word images from a degraded text image, in which character segmentation is difficult because of the existence of touching and broken characters. Visual inter-word constraints available from a text page provides a way to partition word images into smaller parts. We further describe in detail an algorithm under the approach. It is demonstrated that contextual information at the image level can be utilized successfully in character segmentation, an image preprocessing stage for text recognition.

In a highly degraded text image, noise makes several characters touching with each other.
and also makes a character be broken into several pieces. Segmentation points are hard to be located correctly by using the information inside an isolated noisy word image. The method proposed tries to use one word image to segment another word image based on the visual inter-word constraints. A large word image usually more tolerates to noises; therefore using it to segment another word image usually can lead more accurate result.

Although the method can be used by itself for character segmentation. We believe that by combining it with other current character segmentation methods, such as aspect ratio estimation and profile analysis, character segmentation for degraded text will be more accurate and more efficient.

References


Tao Hong received the B.S. degree in Computer Science from Beijing University in 1986, M.S. in Psychology from Beijing University in 1989 and M.S. in Computer Science from the State University of New York at Buffalo in 1992. He is presently a Ph.D. student in computer science department and a research assistant in CEDAR in the State University of New York at Buffalo. His areas of interests includes document recognition and natural language processing.