A Multi-step Method For Text Region Detection In Large Scale Compound Image

Qi Xi, Xing Wu

UM-SJTU Joint Institute

ABSTRACT

Text region detection serves as an important pre-processing step for text extraction and recognition, thus, it has a great significance in the application of image processing technology. However, most of the current methods developed for such a purpose is either time consuming or could only applied to simple background small scale images. In this paper, we present a multi-step method based on connected domain characteristics and edge growth, which accomplish a highly effective detection for large scale compound image with comparable accuracy.

1 Introduction

Text region detection serves as an important pre-processing step for text extraction and recognition, thus, it has a great significance in the application of image processing technology. In this paper, we present a multi-step method based on connected domain characteristics and edge growth, which accomplish a highly effective detection for large scale compound image with comparable accuracy.

Enormous algorithms have been proposed for localization of text regions, many of which are based on non-linear methods such as Markov field and neural networks, requiring bulky computation resources. Gllavata, Ewerth and Freisleben [1] proposed a linear method based on horizontal pixel profile, but it could only be applied to images on which the text regions are well aligned. In this paper, we improve the method in [1] by a double scan procedure, in addition to text stroke filtering and edge growth. B.-F.Wu, C.-C Chiu and Y.-L. Chen[2] proposed extracting characters based on transition pixel ratio, foreground pixel ratio and block size, but this is rather time consuming especially for large scale image. Thus, in his paper, we present a simplified method based on two geometric property of the connected domain: size and smoothness. However, such a simplified method may cause lose of text pixels, the edge growth method proposed by Chen [3], is utilized to compensate the negative effect.

2 Method

Three major steps are taken to detect the text regions in a compound image:
Transforming, Filtering, and Locating. In our multi-step method, the RGB form image is first converted into grey level, then a simplified canny operator is used to detect the edge pixels, transforming the image into binary form. For the filtering part, geometric property is utilized to extract the character strokes, but this may cause misjudgments and lead to split characters, so the edge grow algorithm is manipulated to restore the lost pixels. For locating, horizontal and vertical scan frame the rectangular candidate text regions, according to the number of text pixel. Then each region will adjust its border, expand or contract to the most suitable position. Finally, the adjacent regions will be merged to form a unique region.

2.1 Generation of gray-level image

A majority of images from internet or digital camera are stored in, or compressed from, RGB form. A RGB image is an m-by-n-by-3 data (range:0-255) array that defines red, green, and blue color components for each individual pixel. To reduce redundancy, the raw RGB image are first transformed into grey level image, a monochromatic form which retains the topology property of the original picture. The procedure is as follows: scan the whole image from bottom to top, right to left, to obtain the gray level of each pixel by

\[ Y = (0.257 \times R) + (0.504 \times G) + (0.114 \times B) \]

Where Y is the grey level and R, G, B represent the red, green, and blue component of the corresponding pixel.

2.2 Edge detection with simplified canny operator

Because of the abrupt change in grey level between the text pixels and its neighbor background pixels, edge detection is an effective way to separate the text contour and could reduce a great many useless information. In this paper, we adopt a simplified canny operator proposed by ***. The procedure is as follows:

Consider the grey level of a pixel with its eight adjacent pixels (G0-G8), as shown in Figure 1. We define the gradient at the particular position of pixel 0 as

\[ \text{Grad}(0) = \max\{|G(0) - G(1)|, |G(0) - G(5)|, |G(0) - G(8)|\} \]

The value represents the spatial changing rate of grey level at the point, if this value surpasses a certain threshold (15 in our method), an edge pixel is found. So we scan the whole image pixels by pixels as that in step 1 to find out all the edges. After the detection, the image is converted from grey level form to binary form, with all the edges marked.

2.3 Extraction of character strokes

To separate texts from the background, and remove picture pixels, we adopt a method based on connected domain characteristics. First, connected domains on the binary image from step 2 are labeled; using an equivalent label set method. Then the character strokes are extracted based on the geometric property: size and smoothness. Most of the character strokes are distributed in a certain size interval (range: 80-8000 pixels, experimental value). So size is one of the criteria to select character strokes, the other one is the branch-size ratio, based on the fact that the character strokes usually contains less branches than that of pictorial strokes of the same size. The process is as follows:

Scan the image, left to right, bottom to top, for each marked pixel, acquire the information of its eight neighbor (As shown in Figure 2, P0-P8 represent the marking configuration), consider the value of P5, P6, P7, P8:

All the possible distribution could be sorted to six categories.
1). None of the four pixel is marked, give P0 a new label;
2). One of the four pixels is marked, copy its label to P0;
3). (P5,P6) are marked, copy P5’s label to P0;

...
4). (P6, P7) or (P7, P8) or (P6, P7, P8) are marked, copy P7’s label to P0.;
5). (P5, P7) or (P5, P7, P8) are marked, copy the smaller label from P5 and P7 to P0, put both of the labels to the equivalent label set.
both of the labels to the equivalent label set

**Figure 2.**

The following figures illustrate each of the above situations; note that the mark of P1, P2, P3, P4 are ignored. And all the possibilities of each situation is not listed.

**Figure 3. Different configuration of edge points**

### 2.4 Edge growth

The extraction process in step3 may cause misjudgments and lead to split characters, which will bring negative effect to further locating task. To increase accuracy, restoring the lost pixels is of great necessity. In this paper, we adopted a simplified edge growth algorithm presented by Chen, to connect the discrete strokes. The procedure is as follows:
1. Scan the picture, consider the marking configuration for each marked pixel, still use the concept of connective number Nc as in formula 1.
2. Consider the pixel with connective number of 1, choose its 8-neighbour blank points as candidate new
mark points;
3. Mark the candidate points with the largest gradient level, if there are more than one point, mark the one
with the least gradient level difference with the central pixel.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>0</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 4. Examples of 3*3 window with central pixel connective number is

4. In the original method, the above procedure is repeated until no candidate pixel is found, but as time
goes by, each time only a tiny number of new pixel is marked. So to increase efficiency, we just repeat
such a procedure several times.

2.5 Locating of text region

The binary image from step 4 contains mainly text pixels, so the text region could be detected based on
those information. In our method, First, use a horizontal scan line to obtain the horizontal projection profile of
the whole picture. A line is accepted as a text line if it contains sufficient marked pixel, but in real application,
there might be gaps within a text region, which cause swing in the projection profile, to get a robust method,
we scan five lines each time, and employ two thresholds to mark the start and the end of a text region, both
threshold are determined experimentally and are fixed. Then for each horizontal candidate text region, a
vertical scan line is manipulate to divide the region into vertical sub-regions, using the same criteria as in the
horizontal scan but the threshold is variable related to the height of the region. When all the scan are
finished, the regions too smaller to be a text region are removed.

2.4 Border adjustment

The detect processor in step 5 resulted in rectangular regions parallel to each other. However, in large scale
compound image, the text may lie in several districts which may not perfectly aligned. So the top and bottom
border of each candidate text region should be adjusted. The method is rather simple: Check the line inside
and outside the border, five lines a time, and the border will expend or contract to the line where the
proportion of the average marked pixel surpass a certain threshold.

3 Results

Figure 5. (a) Original Image. (b) Result of extraction. Image size has been adjusted for
convenience.
3 Reference