Line-Touching Character Recognition Based on Dynamic Reference Feature Synthesis

Yoshinobu Hotta, Katsuhito Fujimoto
Fujitsu Laboratories Ltd.
4-1-1 Kamikodanaka, Nakahara-ku, Kawasaki, 211-8588 Japan

ABSTRACT

In recognizing characters written on forms, it often happens that characters overlap with pre-printed form lines. In order to recognize overlapped characters, removal of the line and restoration of the broken character strokes caused by line removal are generally conducted. But it is not easy to restore the broken character strokes accurately especially when the direction of the line and the character stroke are almost same. In this paper, a novel recognition method of line-touching characters without line removal is proposed in order to avoid the difficulty of the stroke restoration problem. A line-touching character is recognized as a whole by matching with reference character features which include a line feature. And the reference features are synthesized dynamically from a character feature and a line feature based on the touching condition of an input line-touching character string. We compared the performance of the proposed method with a conventional method in which a touching line is removed leaving the overlapped character stroke by mathematical morphology. Experimental results show that proposed method can achieves 96.26% character recognition rate whereas the conventional method achieves 92.77%.

Keywords: character recognition, feature synthesis, line-touching, line removal, OCR

1. INTRODUCTION

In recognizing characters written on forms, bills or receipt slips, it often happens that characters overlap with pre-printed form lines. This is caused by deviation of paper position in printing. Fig.1 shows examples of such line-touching character strings.

![Fig. 1 Examples of line-touching character strings](image)

In order to recognize overlapped characters, removal of the line and restoration of the character stroke are generally conducted \(^1\)\(^2\)\(^3\). But it is not easy to restore the broken character stroke accurately in some cases. If a line intersects orthogonally with a character stroke, it is relatively easy to remove the line and restore the broken character stroke. But when the direction of a character stroke and a line are almost same, the restoration is very difficult as shown in Fig.2.

![Fig. 2 Examples of line removal and restoration of the broken character stroke](image)

A patching method is proposed in which broken characters are spliced based on cut points between a line and a character \(^4\). But in this method, characters are sometimes patched incorrectly such as 'b' and 'b'. Also the patching can be applied to characters with simple shape only such as alphanumeric. In other method, touching conditions between a character and its touching line are trained, and the knowledge is used in recognition \(^5\). But this method needs various
types of training samples. Another approach uses several dictionaries each of which is trained by different percentage of cut images. Since this method needs large size of dictionaries, it may be difficult to apply this method to recognition of large category problems such like Chinese character recognition.

To solve these problems, we propose a novel recognition method for line-touching characters. In our method, a line-touching character is recognized as a whole without removal of the line. Thus the difficulty of broken stroke restoration is avoided. Also the reference features of the line-touching character are dynamically synthesized from character features stored in a dictionary and a line feature based on touching condition of the input character strings. Therefore our method is robust in recognition of line-touching characters regardless of line position or line thickness. And the dictionary size is smaller compared with that of conventional method.

In Section 2, an overview of the proposed line-touching character recognition is described. In Section 3, two types of feature synthesis methods, an image-based synthesis and a directional feature-based synthesis, are proposed. Experiments are conducted in Section 4 and the conclusion is set out in Section 5.

2. OVERVIEW OF LINE-TOUCHING CHARACTER RECOGNITION

When a character string image which touches with a line is input, it is binarized and the touching line is temporally removed based on the histogram of black pixels in order to segment each character. Then a character with the touching line is extracted from the original binary image and feature extraction is conducted. At the same time, reference features are synthesized dynamically from character features stored in a dictionary and a line feature based on touching condition of the input string. After that, the extracted line-touching character (LTC) feature is matched with the synthesized reference features. Fig.3 shows the overview of the proposed algorithm.

![Fig. 3 Whole flow of proposed method](image)
3. DYNAMIC REFERENCE FEATURE SYNTHESIS

First, an estimation of a touching line position to a character string is described in Section 3.1. Next two types of LTC (Line-Touching Character) feature synthesis methods are described. They are an image-based synthesis and a directional feature synthesis.

3.1 Estimation of line position

It is necessary to know the position of a touching line to a character string. When the line is temporally removed from an original binary image, a histogram of black pixels is calculated. From this histogram, we can know the height of the character string ($H_s$), the height of the line ($H_L$), and the distance from the top of the string to the line ($L_s$), respectively. In Fig.4 (left), a line touches with a character string at lower position. In Fig.4 (right), a line intersects with a character string.

These parameters are used in LTC feature synthesis.

![Fig. 4 The position of the line](image)

3.2 An image-based LTC feature synthesis

In an image-based LTC (Line-Touching Character) feature synthesis, binary character images of every category are stored in a dictionary. When each line-touching character image in a input string is extracted, the ratio of height between the character and the line is calculated as described above. Based on this ratio, a line image is generated and it is added to each character image in the dictionary. After that, a directional feature of LTC image is extracted from the synthesized LTC image.

Finally the extract LTC feature and the feature extracted from an input character image are matched with each other by Euclidean distance. Fig.5 shows this image-based LTC feature synthesis and the extracted LTC feature from the synthesized image.

![Fig. 5 An image-based LTC feature synthesis](image)

3.3 A directional feature-based LTC feature synthesis

Although the image-based LTC feature synthesis is intuitive, there are several drawbacks. Since LTC image synthesis and feature extraction from the synthesized image are conducted for every category in recognition, its processing time is slow. One of other reasons is the accuracy. In the image-based LTC feature synthesis, mean images of every category are stored in a dictionary and they are matched with an input feature. But only one image per category is stored in the dictionary and the mean image doesn’t represent various type of font images fully.

Therefore other feature synthesis method, a directional feature-based LTC feature synthesis is considered.
In this method, local chain code histograms of character contour \(7\) are used as a feature vector. The rectangular frame enclosing a character is first divided into 13x13 blocks. In each block, a chain code histogram of the character contour is calculated. The feature vector is composed of these local histograms. Since contour orientation is quantized to one of 8 possible values (0, 45, 90, 135, 180, 225, 270, 315 degrees), a histogram in each block has eight components. After the histogram calculation, the 13x13 blocks are down sampled with Gaussian filter into 7x7 blocks. The feature vector has 392 (=7x7x8) elements when all the 49 blocks are included.

In the directional feature-based LTC feature synthesis, the directional feature of each character is stored in the dictionary. And an reference LTC feature is dynamically generated by synthesizing the character feature and a line feature. Fig.6 shows an example of the directional feature-based LTC feature synthesis.

Our concept is that the synthesized feature of Fig.6 can approximate the extracted feature of Fig.5. Thus features of line-touching characters are generated dynamically.

The detailed process of LTC feature synthesis is as follows.

There are 7x7 meshes, each of which includes 8 directional elements. At image level, if a line overlaps with a character stroke, the contour feature of the character cannot be extracted from the masked part of the character stroke (Fig.7). In this case, the vertical or diagonal elements of the character feature are thought to be diminished according to a weight, \(w\ (0 \leq w \leq 1)\). On the other hand, the horizontal features are calculated by another weighted summation between the character element and the line element since the line element itself has horizontal element.

Let \(f_{LTC}^i = (f_{LTC}^{i1}, f_{LTC}^{i2}, \ldots, f_{LTC}^{id})^T\) be a LTC (Line-Touching Character) feature, \(f_C = (f_C^{1}, f_C^{2}, \ldots, f_C^{d})^T\) be a character feature and \(f_L = (f_L^{1}, f_L^{2}, \ldots, f_L^{d})^T\) be a line feature, respectively. \(d\) denotes a feature dimension.

(a) If there are no elements in a mesh of a line feature, the corresponding element of the LTC feature is the same with that of the character feature.

\[ f_{LTC}^{ii} = f_C^{i} \quad (1) \]

(b) If there are non-zero elements in a mesh of a line feature,

(b1) the vertical or diagonal element of the character feature is multiplied by \(w\ (0 \leq w \leq 1)\), and the multiplied value becomes the corresponding element of the LTC feature. The weight is determined by experiment.
The horizontal element of the LTC feature is calculated by averaging the corresponding elements of character feature and the line feature.

\[ f_i^{LTC} = f_i^C \ast w \]  

After the LTC feature synthesis, the synthesized feature is matched with the feature extracted from an input character image by Euclidean distance.

4. EXPERIMENT

4.1 A weight decision in the directional feature-based LTC feature synthesis

First we generated line-touching character strings which includes 0 to 9 characters by connecting character images. Then a line is added afterward changing its position. The position is set to three types such as upper, middle, lower. Thus we got 94 strings including 940 characters. Fig.8 shows some of the generated strings using lots of font patterns.

![Generated line-touching character strings](image)

In order to determine the weight value in Section 3.3, we changed it from 0.1 to 1.0 and calculated the character recognition rate. The experimental result is shown in Fig.9.

![Character recognition rate according to the weight value](image)
When the weight value is 0.5, the character recognition rate is highest. If the weight value is smaller or larger than that value, the rate gradually drops. Thus we selected the weight value, 0.5, as an optimal one.

The ratio of height between a line and a character in the generated strings is set to 1:7. If the ratio changes, the optimal weight value may change. Ideally, the optimal value should be decided according to the ratio in an input line-touching character string.

### 4.2 Comparison Test

Next, we collected 83 numeral string images from real forms and used them as test data. In order to compare the performance of the proposed method with conventional methods, we adopted two conventional methods. In the first method, a touching line is just removed before character recognition. In the second method, a touching line is removed and part of an overlapped character stroke is left by using opening and closing operation of mathematical morphology. As for proposed methods, both the image-based feature synthesis and the directional feature-based feature synthesis are tested. In the image-based feature synthesis, about 100 binary font images per category are averaged to a grayscale image. After binarization of the grayscale image, the binarized image is used in synthesis. In the directional feature-based synthesis, feature extraction is conducted for about 100 binary font images per category and the average feature is used in synthesis. The weight value is set to 0.5.

The experimental result is shown in Table.1 and Fig.10.

<table>
<thead>
<tr>
<th></th>
<th>Conventional 1 (Line removal)</th>
<th>Conventional 2 (Morphology)</th>
<th>Image-based</th>
<th>Directional feature-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upper (23 strings, 103 chars.)</td>
<td>82.52</td>
<td>89.32</td>
<td>90.29</td>
<td>92.23</td>
</tr>
<tr>
<td>Middle (36 strings, 187 chars.)</td>
<td>93.58</td>
<td>94.12</td>
<td>94.12</td>
<td>97.86</td>
</tr>
<tr>
<td>Lower (24 strings, 111 chars.)</td>
<td>98.20</td>
<td>93.69</td>
<td>95.50</td>
<td>97.30</td>
</tr>
<tr>
<td>Total (83 strings, 401 chars.)</td>
<td><strong>92.02</strong></td>
<td><strong>92.77</strong></td>
<td><strong>93.52</strong></td>
<td><strong>96.26</strong></td>
</tr>
</tbody>
</table>

**Fig. 10** Comparison of character recognition rate
Compared with the conventional methods, the performance of our proposed methods are better in total. Furthermore the performance of the directional feature-based synthesis is much better than that of the image-based synthesis. Examples of correctly recognized character strings are shown in Fig.11. Even when the line and the character stroke overlap with same direction, they are recognized correctly. Some characters are mis-recognized because of the degradation of characters or similar shape problems caused by feature synthesis method as shown in Fig.12.

![Fig. 11 Examples of correctly recognized character strings](image1)

According to the detailed investigation, the effectiveness of each method differs depending on the line position and its touching character category. As shown in Fig.12, if a line touches with “7” at lower position, the line removal method achieves best performance among all the methods. Since the shape of line-touched “7” looks like “2”, it is not easy to discriminate it from line-touched “2” by our proposed methods. Also if a line touches with “0” at middle position, the morphology-based method achieves best performance since the interpolation of the character stroke of “0” after line removal works well, whereas our proposed methods cannot discriminate it from line-touched “8” in some fonts.

On the other hand, if a line touches with “2” at lower position, the line removal method or the morphology-based method cannot deal with it properly. Since the occurrence of “2” in the test data is few, the line removal method seems to achieve best performance for lower touching string data in Fig.10.

5. CONCLUSION

In this paper, a line-touching character (LTC) recognition method based on dynamic reference feature synthesis is proposed. Whereas conventional methods try to remove the overlapping line on a character string and restore the broken character stroke, our method recognize the line-touching character as a whole without line removal. And the reference features of line-touching characters are generated dynamically from character features stored in a dictionary and a line feature.

Two types of LTC feature synthesis methods are introduced. They are an image-based synthesis and a directional feature synthesis.

Experimental result shows that the directional feature-based synthesis method can achieve 96.26 % character recognition rate for 83 strings including 401 characters whereas conventional method of morphology-based method achieves 92.77%.
Our future work is to design more precise feature synthesis model and improve the performance and then combine it with conventional methods according to a touching line position and the touched character category. The proposed method can be applied to line-touching character recognition of alphanumeric, Chinese characters and handwritten characters if only the character feature type are same. So we will further conduct various experiments to verify the possibility of this method.

REFERENCES