A FAST ALGORITHM OF ADDRESS LINES EXTRACTION ON COMPLEX CHINESE MAIL PIECES

Tong Liu, Xiaoqing Ding, Qiang Fu
State Key Lab of Intelligent Technology and Systems
Tsinghua University
Beijing, P.R.China
{liutong,dxq, fuq}@ocrserv.ee.tsinghua.edu.cn

Zheng Ren
Siemens AG
Buecklestrasse 1-5
D-78467 Konstanz, Germany
zheng.ren@siemens.com

ABSTRACT
A fast and efficient method is presented to extract address lines on both machine printed and handwritten Chinese mail envelopes. The algorithm is based on a bottom-up approach. First, we select out text blocks from connected components (CCs) and immediately group the text blocks into the initial lines. Then, the average text block features are computed to validate the initial text lines and guide an iterative split and merge process. Lines are split by merging the text CCs in detail according to criteria for similarity and consistency of neighborhood text blocks. Particularly, some non-text blocks within the lines are recovered if they are similar with other text blocks. A skew detection and, accordingly, deskew step is followed. We have tested the performance of our methods on a large mail sample test deck with different categories of envelopes, and an obvious improvement both on accuracy and on computation time could be achieved compared to our previous system.

KEY WORDS
Chinese mail address lines, connected components Analysis, similarity, split and merge

1 Introduction
Mail address lines contain much more information than just postcode to facilitate postal automation. With the availability of OCR technology for recognizing both handwritten and machine printed Chinese characters with high accuracy, it is possible now to explore full address information on mail envelopes to improve the effectiveness, and efficiency of automated mail sorting machines [1]. Xue J., Ding X., Liu C., et al.[2] have taken the first research step on Chinese mail image analysis. However, location of the address lines on Chinese mail pieces still faces difficulties: different kinds of mail pieces, complexity of Chinese characters, variance of position and inter-line distance, influence of skew and other image/graphic blocks, etc. Our purpose is to extract all address lines on complex Chinese mail pieces to feed into our address reading machine.

Addresses location in mail images is a prerequisite for postal automation systems with recognition functionality. Several methods have been advanced to locate address blocks in envelope. These methods can be classified into three different approaches based on geometric features, expert systems, and texture analysis. Adrian, P.W., Hong Y.[3] presented a method for automatically locating address candidate address blocks directly. This system used high-pass filter and a serial of constraints to select the best address block. Yeh, P. et al. [4] utilized mail geometric features and Laplacian operator. Based on the gray level connected component analysis, different clusters of components were produced and the best block was chosen as the real address block. Wang [5] developed an expert system to locate the address block. Some rules are obtained from the database. With this rules and geometric features, some candidate address block were obtained and the best one was chosen. Jain, A.K., Bhattacharjee, S.K. [6] proposed a method using texture analysis. Based on Gabor band-pass filter of two frequencies and four orientations, an input image was decomposed into a number of filtered images. Then three regions consisting of text, non-text and boundaries between text and non-text were identified. Finally the text cluster was selected and the address block formed.

In the current paper, a new bottom-up approach to extract candidate address lines is proposed. Unlike others, we adopt the local geometric features which are updated with a merging process. Moreover, the merging process is carried out in accordance to similarity and consistency investigation of neighbour text blocks, under guidance of average text block features obtained from initial lines. Finally, we perform a validation algorithm to remove the obviously non-text lines and recover the blocks in the candidate lines.

The remainder of this paper is organized as follows. First in section 2, we present the preprocessing task. In section 3 our address lines location algorithm is described in detail. Section 4 compares and analyzes our method. In section 5 conclusions are drawn finally.

2 Preprocessing
A flow chart of our address lines extraction algorithm is shown in Fig. 1. We divide mail piece layouts into four categories: handwritten, normal machine printed, envelope with plastic window cover, envelope with printed label, as shown in Fig. 2.
2.1 Special Processing

In order to facilitate the address lines extraction, we first perform a preprocessing step. A median filter is first adopted to remove the isolated noise introduced in the scanning process. A morphological close operation both in horizontal and vertical direction is followed to enhance the degraded images. Then we remove the bar code and lines around the address labels and horizontal lines printed on Chinese mail. Furthermore, the high resolution is not needed in the address lines extraction step. So the resolution of the original image is reduced by a factor of 4 both in the width and in the height.

2.2 Connected Components Classification

There are many methods for connected components extraction. We use a method scanning from top to bottom and from left to right to merge the components and generate new ones. After that, according to predefined rules, the CCs are classified into 4 categories $CC_{\text{noise}}, CC_{\text{graphic}}, CC_{\text{image}}, CC_{\text{text}}$, as shown in Fig. 3. Only the $CC_{\text{text}}$ is used to form address lines.

Let the circumscribed rectangle $(x, y, w, h)$ $(x, y)$ is the top-left point of the $CC$, while $(w, h)$ denote the width and height of the $CC$ denote the connected components. In the first step, if the width and height of the $CC$ is less than $T_w, T_h$, its size is too small, and therefore it is labeled as $CC_{\text{noise}}$. Then if $width(CC)/height(CC) > R_w$, 

![Figure 1. The flowchart of our address lines extraction](image1)

![Figure 2. Four types of real envelope samples](image2)
3 Address line extraction

Text lines extraction is a common problem in many application systems, such as check, tables, postal automation reading systems, and many others in complex layouts. As to Chinese mail pieces, the Chinese characters have some unique features: 1. Most of the Chinese characters contain more than one connected components, and the stroke distributions are more complex than western letters. 2. The writing styles of the address lines and the distribution of different blocks vary greatly from western mail pieces. Thus, these methods are not suited and applicable to Chinese reading machine.

We propose a fast and effective method based on similarity of the neighboring connected components. After that, a more complex check process is performed to find out the most consistent address lines. The purpose of our analysis is to find all relevant address lines including postcode line, receiver’s street address line and corporation name line. To this aim we have designed a candidate-dependent system which is basically different from our old system [2]. The detail of our algorithm is listed in the following subsections.

3.1 Initial lines extraction

After preprocessing in section 2, a serial of text blocks are selected out to form text lines. We denote them as \( C = \{ C_1, C_2, \ldots, C_N \} \), where \( N \) is the total number of text blocks, and every \( C \) represent the circumscribed rectangle called Bounding Box. Then we generate initial lines only through Bounding Box horizontal projection. The connected components between the valley of the histogram are cumulated as one line. A number of initial text lines are generated immediately, denoted as \( L = \{ L_1, L_2, \ldots, L_M \} \), where \( M \) is the number of initial lines. \( L_i = \{ C_{i,k} \} k = 1, \ldots, N \) is a horizontal text line composed of \( N \) neighboring connected components.

In the real situation, the image of the envelope is not so clean. Bounding Box horizontal projection may be influenced by some unexpected noise, such as the intersection between lines and some noise block within the text lines. These could cause the lines generated in the first step containing more than one address line. So, a validation of the initial lines is performed. Only the lines that are checked as multi-lines are split and merged, while the single lines remain unchanged. This approach could save a lot of time for single lines which do not need split and merge process.

3.2 Line split and merge

When an initial line is judged as containing multiple lines, we implement a successive split and merge operation according to the neighboring connected components’ similarity. First of all, we compute the features of connected components (CCs) within the multiple lines. The text line information data structure is shown as follows.

Text line low level info data structure

```c
typedef struct {
    RECT rcRectSrc;
    // Circumscribed rectangle of the line
    int nAvgWidth;
    // Average CC block width
    int nAvgHeight;
    // Average CC block height
    int nAvgArea;
    // Average CC block area
} LW_TXTLINEINFO;
```

Text line high level info data structure

```c
typedef struct {
    int nCharNum;
    // the number of character in the line
    int nCharAvgHeight;
    // the average height of the characters
    int nCharAvgInterval;
    // the average interval of the characters
    int nCharAvgWidth;
    // the average width of the characters
} HL_TXTLINEINFO;
```
and $O_h(a,b)$ mean the vertical and horizontal distance of non-overlapped blocks, where $O_v(a,b)$ and $O_h(a,b)$ denote the vertical and horizontal distance of overlapped blocks.

With such statistical split block features, we could supervise the following split and merge step.

1. Given the initial line $L_i = \{C_{i,k}\}$ which needs to be split and merged. The circumscribed rectangle $(x,y,w,h)$ is represented by $C_{i,k}$. Suppose $C_{ia}$ and $C_{ib}$ denote two text blocks with the line $L_i$. $D_v(a,b)$ and $D_h(a,b)$ denote the vertical and horizontal distance of the two non-overlapped blocks respectively, also $O_v(a,b)$ and $O_h(a,b)$ denote the vertical and horizontal distance between two overlapped lines respectively. The relations between the CC are shown as Fig. 4. Then assign a flag $nLine$ to each $C_{i,k}$, when the $C_{i,k}$ is grouped as a multiple lines, $nLine = 1$, else, $nLine = -1$.

2. Find the topmost text block $CC_0$ within the line $L_i$. Then group it to the temporary line $L_{temp} = \{CC_0\}$, and let the $CC_0\{nLine = 1\}$.

3. With starting block $CC_0$, find the most neighboring text block and merge into $L_{temp}$ repeatedly until all text blocks belong to the some lines. Supposing that the current text block is $CC_{i,k}$, which is already merged into $L_{temp} = L_{temp} + CC_{i,k}$, the next text block $CC_{i,k+1}$ could be merged into $L_{temp}$, if it satisfies the following rules.

$$CC_{i,k+1} = \min_{CC \in L_i} \{D_h(CC_{i,k},CC_{i,k+1}),$$
$$O_h(CC_{i,k},CC_{i,k+1})\}$$

$$O_v(CC_{i,k+1},L_{temp}) > T_v$$

$$Height(CC_{i,k+1}) < 2* nAvgHeight$$

$$D_h(CC_{i,k},CC_{i,k+1}) < T_h$$

Where $T_v$ is acquired by prior knowledge and $T_h$ is determined by the average interval of characters in the initial lines (nCharAvgInterval). If $CC_{i,k+1}$ satisfy the similarity requirement, merge it into the line $L_{temp} = L_{temp} + \{CC_{i,k+1}\}$, and set the $CC_{i,k+1}\{nLine = 1\}$. Iterate until no text block with the attribute $nLine = -1$.

4. Repeat the above step, we will obtain a serial of temporary text lines $L_{temp} = l_1,l_2,\ldots,l_k$.

5. Check all the connected components within the temporary lines$\{l_1,l_2,\ldots,l_k\}$. If one non-text block has some common features such as width, height, width/height, height/width, etc., we will recover it as text block.

3.3 Skew Detection and Deskew

We first detect the skew angle of the extracted lines, and then apply a deskew algorithm to adjust the line to a horizontal direction in order to enhance the accuracy of the recognition. We adopt the robust and fast skew detection algorithm [10]. Firstly extract the centroids of connected components using graphic data structure. Then apply a hierarchical Hough transform to selected centroids. The skew angle corresponds to the location of highest peak in the Hough space.

4 Experimental Results

To evaluate our performance of our proposed method, we collect 10,000 different kinds of mail envelope images scanned at the resolution of 212 dpi. The test samples include 4,000 handwritten mail images, 5,000 normal machine printed mail images with different character size and inter-line distance, 500 mail images pasted with printed label, 500 mail images with plastic window cover. If post-code line, destination address lines, and corporation name lines, are correctly extracted, we consider it as a correct extraction. Some address lines extraction results are shown in Fig. 2.

Table 1 shows the performance of the proposed method in this paper. A correct rate of 99.2% could be achieved on normal machine print mail images. While test on mail images with plastic window cover, the correct rate decreases to 97.6%, because of influence of the plastic cover. Mail images pasted with print label get the lowest recognition rate 92.2%, because the position and pattern of the address block vary greatly. On handwritten mail images, a correct rate of 95.08% could be achieved. As shown in Table.1, compared to our old system, we could see our new method could improve the performance on printed envelopes notably, especially on printed envelopes with plastic window cover and printed label. Envelopes with plastic cover and printed label often carry more noise block within the address area and various skew angles. Our old system does not consider the similarity and consistency of CCs in the merging process. While our new bottom-up approaches could tolerate skew angles within a certain range, and performs much better on printed samples. Overall, our new algorithm achieves a better performance, especially on machine printed envelopes which increase greatly.

From Table 2, we could see the average processing time of each part per envelop, which was tested on the Pen-
Table 1. Performance of address lines extraction

<table>
<thead>
<tr>
<th></th>
<th>Total mail</th>
<th>Our old system</th>
<th>Our new method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Wrongly extracted</td>
<td>Correct rate</td>
</tr>
<tr>
<td>Mail kind</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Machine print</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>normal printed</td>
<td>5000</td>
<td>128</td>
<td>97.44%</td>
</tr>
<tr>
<td>plastic window cover</td>
<td>500</td>
<td>122</td>
<td>75.6%</td>
</tr>
<tr>
<td>printed label</td>
<td>500</td>
<td>88</td>
<td>83.4%</td>
</tr>
<tr>
<td>Handwritten</td>
<td>4000</td>
<td>232</td>
<td>94.2%</td>
</tr>
<tr>
<td>Total</td>
<td>10000</td>
<td>565</td>
<td>94.35%</td>
</tr>
</tbody>
</table>

Table 2. Average processing time per mail image

<table>
<thead>
<tr>
<th></th>
<th>Preprocessing</th>
<th>CC Extraction</th>
<th>CC Classification</th>
<th>Line Extraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average time(ms)</td>
<td>227</td>
<td>114</td>
<td>13</td>
<td>32</td>
</tr>
</tbody>
</table>

tium 2.8G PC. Such short processing time can satisfy the real-time processing requirement for our system.

We divide the errors in address lines extraction into two types, as shown in Table 3. The first type is partial extracted, that is, some components that belong to the address lines are lost. While the second type error is over extracted, that is, we group some undesired non-text components into the address lines. The first type error is caused by some noise blocks which intersect into the address lines, and some Chinese radicals occasionally are separated out to form a new line. While the second type error arise from merging some undesired blocks or overlap of handwritten address lines. Compared to the second type, the first is vital, because the over extract error might be adjust by the successive part. So it is important to recover the text blocks which is discarded in the merging process.

Table 3. Error analysis

<table>
<thead>
<tr>
<th>Total error</th>
<th>Partial extracted</th>
<th>Over extracted</th>
</tr>
</thead>
<tbody>
<tr>
<td>284</td>
<td>93</td>
<td>191</td>
</tr>
<tr>
<td>2.84%</td>
<td>0.93%</td>
<td>1.91%</td>
</tr>
</tbody>
</table>

5 Conclusion

This paper presents a fast and efficient method to extract the address lines in Chinese mail images. A new bottom-up algorithm is proposed. We have successfully embedded the method to an address recognition system.

Different to other reported methods, we classify the connected components into four clusters, merge text blocks into lines based on their similarity, and recover the text blocks that are not merged. Furthermore, we adopt a rough initial lines extraction and iterative split and merge process, which could save a lot of time. We test our algorithm on various kinds of mail images including both handwritten, and machine print. A total correct rate of 97.16% could be achieved.

However, further research is needed to solve the difficulty of more complex mail pieces and envelopes written with different styles. We should also need to devise a more robust and detailed line validation algorithm faced with the complex handwritten envelopes. Moreover, in order to make the real time processing faster, we still need to reduce the processing time.

Acknowledgements

This work has been funded by Siemens AG under contract number 20030829-24022SI202.

References

[7] V. Govindaraju, S. Tulyakov, Postal address block location by contour clustering, Proc. 7th Int. Conf. on Doc-