Handwritten ZIP Code Recognition
Using Lexicon Free Word Recognition Algorithm

F. Kimura†, Y. Miyake‡, and M. Shridhar+++,

† Faculty of Engineering, Mie University
1515 Kamihama Tsu 514, JAPAN
‡ ECE Dept., University of Michigan-Dearborn
Dearborn, MI 48128-1491, USA

Abstract
This paper describes a new approach to ZIP code recognition using a word recognition algorithm, where a numeral string is recognized as a word. This paper also describes an end to end ZIP code recognition system consisting of tilt/slant correction, line segmentation, word segmentation, ZIP code location, and correction is more difficult for numeral string than for
words. To meet this requirement, two different character classifiers are used in the internal segmentation and the post numeral recognition respectively. This paper also describes an end to end ZIP code recognition system consisting of tilt/slant correction, line segmentation, word
segmentation, ZIP code location, as well as the ZIP code recognition. The segmentation of lines and words is in many ways similar to the character segmentation except for the different levels. Accurate numeral string segmentation is often very difficult (especially if the numeral strings are strongly connected) but an essential task for ZIP code recognition. In the proposed ZIP code recognition system, final segmentation and recognition are concurrently determined using word recognition techniques, while the line segmentation is pre-determined prior to the recognition task.

Evaluation tests are performed using address block image samples collected from United States mail pieces. The results of isolated numeral recognition, manually extracted ZIP code recognition, and end to end ZIP code recognition are presented to show and discuss the advantage of the word recognition based numeral string recognition.

1 Introduction
This paper describes a new approach to ZIP code recognition using a word recognition algorithm, where a numeral string is recognized as a word. Recognition of numeral string is more difficult than isolated numeral recognition because of the segmentation problem. Numerals are frequently touching, connected or broken into segments due to variations in writing style, writing equipment, and the digitizing process.

Conventional numeral string recognition is essentially based on external segmentation, which is performed prior to numeral recognition. For example, if the numeral string has less components than expected, the number of numerals in each component is estimated based on the width, and multi-numeral components are split [1, 2]. However, the correct detection of multi-numeral components is not an easy task due to variations in numeral width, broken numerals, noise, and other unavoidable factors. This segmentation problem is common to word recognition and can be resolved by a lexicon free word recognition algorithm with internal character segmentation. The internal character segmentation (or segmentation-recognition) provides global optimal segmentation with regard to the entire character recognition in a word.

For internal segmentation, high speed and high accuracy numeral recognition is essential because error detection and correction is more difficult for numeral string than for words. To meet this requirement, two different character classifiers are used in the internal segmentation and the post numeral recognition respectively. This paper also describes an end to end ZIP code recognition system consisting of tilt/slant correction, line segmentation, word...
defined so that the closer the height of the component is to the estimated character height, the larger the weight is. The number K of clusters is assumed to range from one to six. Among the six sets of clusters, those which have poor separability are discarded. Those which do not satisfy spatial constraints required by valid address lines are also discarded. The remaining clustering with maximum number of clusters (lines) is selected and the components are classified into lines. Multiple line components occupying vertical extent of more than two lines are split at the line boundary (Fig. 2). As the separability measure for clusters, the F-ratio for distributions of \((y_{\text{min}}, y_{\text{max}})\) is used.

2.3 Slant correction
The slant correction is performed to each line. The slant estimation algorithm utilizes the chain code of border pixels [5]. The average slant is estimated by

\[
\theta = \tan^{-1}\left(\frac{n_1 + n_2 + n_3}{n_1 - n_3}\right)
\]

where \(n_i\) is the number of chain elements at an angle of \(i \times 45^\circ\) (/ or \(l\) or \(\backslash\)). Shear transformation is then applied to correct the slant of estimated magnitude.

2.4 Word pre-segmentation
The slant corrected address lines are supplied to the word pre-segmentation algorithm. Words are assumed to be separated by a space, a comma, and a period. The space detection algorithm detects the spaces by classifying the gaps between the character segments into 'between-field gap' and 'within-field gap' respectively. If the gap is wider than a threshold, the gap is the 'between-field gap', otherwise the 'within-field gap'. The threshold is selected by applying Otsu's method [6] to the distribution of the gap width for address lines. This space detection algorithm is employed both for handprinted and handwritten (cursive) address blocks.

The word pre-segmentation is assumed to yield over segmented word images. When under segmentation is predicted, the line is further word segmented. If the number of words in a line is less than three and the maximum estimated length of a word is greater than six, the word is divided at the maximum within-field gap. This procedure is repeated (at most twice) until no further subdivision is made.

3. ZIP code recognition subsystem
3.1 ZIP code location and recognition
ZIP code is first assumed to be at the last field of the last line. If the likelihood of the detected ZIP code is less than a threshold, up to two preceding lines are assumed successively to be the ZIP code line until the ZIP code with sufficient likelihood is detected. In actual word pre-segmented images, ZIP code fields often split and divided into several pieces, which have to be merged again into a field. This problem is resolved through multiple use of word recognition algorithm to a set of successive word primitives. The word recognition algorithm employs a lexicon free word matching (described in 4).
4. Lexicon free word recognition algorithm

The authors proposed an algorithm for recognizing unconstrainedly handwritten words [5, 7]. It is a lexicon directed word recognition algorithm with internal character segmentation. The internal character segmentation (or segmentation-recognition) provides global optimal segmentation regarding the entire character recognition in a word. In the lexicon directed algorithm, the ASCII lexicon of possible words is utilized in the optimization process using dynamic programming to incorporate contextual information. Given a lexicon word, the primitive segments obtained from an input word image are merged and matched against a letter in the lexicon word so that the average character likelihood is maximized. The word matching process is repeated for all lexicon words and the lexicon words are sorted and ranked according to the associated average character likelihood.

A lexicon free word recognition algorithm is easily obtained from the lexicon directed algorithm by a simple modification. In the lexicon directed word matching, character likelihood of corresponding primitive segments is calculated for a single letter in a specified position of a lexicon word. While in the lexicon free word matching, the character likelihood for all letters, e.g. 0 to 9 in ZIP code recognition, are calculated and the maximum value and the associated letter are determined. The word matching process is applied only once for an input word.

5. Character recognition

5.1 High speed character recognition

Local chain code histograms of character contour are used as a feature vector (Fig.3). The rectangular frame enclosing a character is first divided into 7 x 7 blocks. In each block, a chain code histogram of character contour is calculated. The feature vector is composed of these local histograms. Since contour orientation is quantized to one of 4 possible values (0° or 90°, 45° or 135° or 180° or 270°), a histogram in each block has four components. After the histogram calculation, the 7 x 7 blocks are down sampled with Gaussian filter into 4 x 4 blocks. Thus the feature vector has 64 elements when all the 16 blocks are included.

One critical point in internal segmentation, using dynamic programming is the speed of feature extraction, because the correct segmentation points have to be determined in optimization process with respect to the total likelihood of entire characters. The use of cumulative orientation histogram enables one to realize high speed feature extraction. Border following and chain coding are performed only once to an input word image, and the orientation histogram in a block is calculated by a small number of arithmetic operations in the internal segmentation process. For more detail of high speed feature extraction see [5].

Character likelihood is calculated using a modified quadratic discriminant function which is less sensitive to the estimation error of the covariance matrix and requires less computation time and storage than the ordinary quadratic discriminant function [8].

5.2 High accuracy character recognition

The accuracy of character classifier employed in internal segmentation is restricted due to the requirement on the computation time. However, once the internal character segmentation is accomplished, a high accuracy character classifier can be applied to each segmented character to obtain a more accurate character likelihood with relatively small additional processing time. As long as the character segmentation is correct and the accuracy of the post classifier is higher, word recognition rate is improved.

A high accuracy character classifier is implemented using high dimensional feature vector of size 400. The 400-D feature vector is obtained in the similar way as for 64-D feature vector. The number of blocks is initially 9 x 9 and down sampled to 5 x 5. The quantization level is 16 directions instead of 4 orientations. To obtain 16 directions, Gaussian filter and Roberts filter are applied to a character image to obtain a gradient image. The tangent of the gradient is quantized into 16 directions and the strength of the gradient is accumulated in each direction in each block. For more details of high dimensional feature extraction see [9]. Gradient-based feature is also described and studied in [10], where directions of gradient are quantized into 12 bits with an OR operation to obtain binary-valued vector.

6. Performance evaluation

6.1 Performance evaluation of numeral classifier

Table 1 and 2 show the performance of high speed and high accuracy numeral classifiers respectively. The top correct recognition rates were 98.30% and 99.21% respectively. Table 3 shows error v.s. reject rates of high accuracy numeral classifier at various operating points. The table shows how the top-choice error rate can be reduced by rejecting numerals whose 1st- and 2nd-ranked numeral do not meet a confidence criteria. Each operating point is determined by two threshold values \( t_1 \) and \( t_2 \). An input numeral image is rejected if the negative log-
likelihood (dissimilarity) of the 1st-ranked numeral is greater than \( t_1 \), or the difference between the 1st- and 2nd-ranked numerals is less than \( t_2 \). For example, at \((t_1, t_2) = (120.0, 96.0)\), the error rate was 0.18\% with 2.58\% rejection.

In this experiment, six numeral sample sets generated from "Bd", and "ba" address block image data were used for design and test. The name and the sample size are shown in Table 4. Among these sample sets, bha_01 was used for test, and the rest of the sets for design. The total size of design sample is 19,488.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Correct rates</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>98.30 (5495)</td>
</tr>
<tr>
<td>2</td>
<td>99.55 (5586)</td>
</tr>
<tr>
<td>5</td>
<td>100.00 (5590)</td>
</tr>
</tbody>
</table>

Table 1: Cumulative correct rates of high speed character recognition

Table 2: Cumulative correct rates of high accuracy character recognition

Table 3: Error v.s. reject of high accuracy character recognition

Table 4: Used numeral sample set

Table 5: Cumulative correct rate of ZIP code recognition (manual field extraction, without high accuracy character recognition)

Table 6: Cumulative correct rate of ZIP code recognition (manual field extraction, with high accuracy character recognition)

Table 7: Error v.s. reject of ZIP code recognition (manual field extraction, with high accuracy character recognition, with validity check using City/State postal directory)

### 6.2. Performance evaluation of ZIP code recognition (manual field extraction)

The performance of ZIP code recognition by lexicon free word matching was evaluated by recognition experiments on manually extracted ZIP Code data (bha_01). The word matching is applied with specified word length, 5 or 10 (including hyphen for 9 digit zip code). If the upper bound of word length estimation is less than 10, input word is assumed to be five digit ZIP code, and if the lower bound of word length estimation is greater than 5, the input word is assumed to be nine digit ZIP code. Otherwise it is assumed either five or nine digit ZIP code. The performance of ZIP Code recognition with and without high accuracy character recognition were evaluated and compared experimentally. Table 5 to 7 show the result for bha_01 ZIP Code data. The top correct recognition rate with and without high accuracy character recognition was 90.38\% and 85.90\%. The main causes of recognition error were numeral recognition (4\%), pre segmentation (3\%), and under line (2\%). Fig.4 shows examples of correctly recognized ZIP code. Table 7 summarize the tradeoff between reject and top-choice error rates at different operating points when high accuracy character classifier is employed. For example, at \((t_1, t_2) = (40.0, 19.0)\), the error rate was 0.61\% with 21.79\% rejection. This result is considerably better than the result of 2.56\% error with 31.25\% rejection, when only the high speed character classifier is employed. Unused U.S. ZIP code is rejected through the city/state Postal directory search.

6.3 Performance evaluation of end to end ZIP code recognition

Tables 8 and 9 show the result of end to end ZIP code recognition. The performance was evaluated using "ba" address block samples. Among the sample images from bha_0000 to bha_5999, 3540 images having the truth value of ZIP code were used in this test. Table 8 summarizes the cumulative correct rates of ZIP code recognition. The top correct recognition rate was 82.20\%. Table 9 summarizes the tradeoff between reject and top-choice error rates at different operating points. For example, at \((t_1, t_2) = (24.0, 24.0)\), the error rate was 1.97\% with 28.28\% rejection. Unused U.S. ZIP code is rejected through the city/state Postal directory search.
Table 8. Cumulative correct rates of ZIP code recognition

<table>
<thead>
<tr>
<th>Rank</th>
<th>Correct rates %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>82.20 (2910)</td>
</tr>
<tr>
<td>2</td>
<td>85.35 (3021)</td>
</tr>
<tr>
<td>3</td>
<td>86.61 (3066)</td>
</tr>
<tr>
<td>Total</td>
<td>100.00 (3540)</td>
</tr>
</tbody>
</table>

Table 9. Error v.s. reject of ZIP code recognition (with validity check using City/State postal directory)

<table>
<thead>
<tr>
<th>Reject %</th>
<th>Error %</th>
<th>Correct %</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.21 (326)</td>
<td>9.96 (320)</td>
<td>90.04 (2894)</td>
</tr>
<tr>
<td>17.63 (624)</td>
<td>4.87 (142)</td>
<td>95.13 (2774)</td>
</tr>
<tr>
<td>28.28 (1001)</td>
<td>1.97 (50)</td>
<td>98.03 (2489)</td>
</tr>
<tr>
<td>50.08 (1773)</td>
<td>0.91 (16)</td>
<td>99.09 (1751)</td>
</tr>
</tbody>
</table>

Fig. 4 Examples of correctly recognized ZIP code

7. Summary and discussion

In this paper, a ZIP code recognition system was described.

The result of evaluation test is summarized as follows.
1) The accuracy of high speed and high accuracy numeral recognition was 98.30% and 99.21% respectively.
2) The high dimensional feature vector derived from gradient image gave remarkably higher accuracy than the low dimensional feature vector. To the authors best knowledge, the accuracy over 99% for U.S. ZIP code numerals has not been reported [10, 11].
3) The accuracy of manually extracted ZIP code recognition with and without high accuracy numeral recognition was 90.38% and 85.90% respectively.
4) The accuracy over 99% for U.S. ZIP code numerals has not been reported [10, 11].
5) The error rate of end to end ZIP code recognition was 1.97% with 71.72% accept rate.
6) The high dimensional feature vector derived from gradient image gave remarkably higher accuracy than the low dimensional feature vector. To the authors best knowledge, the accuracy over 99% for U.S. ZIP code numerals has not been reported [10, 11].
7) The result shows that the high accuracy post numeral recognition is quite effective for word recognition based numeral recognition. This result is also considerably better than that of conventional method [2].
8) The error rate of end to end ZIP code recognition was 1.97% with 71.72% accept rate.
9) The result is very encouraging when the wide variety of handwritten addresses is taken into consideration. The performance is significant and comparable with the latest results reported by other investigators [12].
10) There are many remaining future works for improving total performance. Some of them are:

1) Extension of word matching capability in the presence of noise components,
2) Improvement of pre-segmentation algorithm for numeral strings,
3) Further improvement of numeral recognition accuracy,
4) Integration of City/State name recognition in ZIP code recognition subsystem, and
5) Program optimization for high speed processing.

Acknowledgments

The authors would like to acknowledge the support of United States Postal Service for the research described in this paper. In particular, the authors acknowledge Dr. Binh Phan of USPS and Dr. John Tan of Arthur D. Little, Inc. for their comments and many useful suggestions.

References