Handwritten Japanese Address Recognition Technique Based on Improved Phased Search of Candidate Rectangle Lattice

Hidehisa NAKAYAMA†, Masato SUZUKI‡, Nei KATO†, and Yoshiaki NEMOTO†

† Graduate School of Information Sciences, Tohoku University
aza-Aoba 05, Aramaki, Aoba-ku, Sendai-shi, Miyagi, 980–8579 Japan
‡ Department of Computer Science, Tokyo National College of Technology
Kunugida-machi 1220-2, Hachioji-shi, Tokyo, 193–0997 Japan

ABSTRACT

In the field of handwritten Japanese address recognition, it is common to recognize place-name strings from place-name images. However, in practice, it is necessary to recognize the place-name strings from address images. Therefore, we have proposed the post-processing system, which checks the list of the place-name strings in two-stages for recognizing the place-name images. In this paper, we propose a new technique based on phased search of candidate rectangle lattice, and improve the technique with the detection of key-characters for final output. Applying our proposal to the IPTP 1840 image data of address strings, the results of experiments clearly show the efficiency of our system in handwritten Japanese address recognition.

Keywords
Handwritten Japanese Address Recognition, Address Image, Candidate Rectangle Lattice, Key-Characters, Phased Search and Post-processing.

1. INTRODUCTION

A Japanese address string consists of two elements, which are a place-name string and a place-number string. Handwritten Japanese address recognition sets a goal to read the place-name string from its address-name image on the postcard without referring to the zip code. In the general system, an input image is segmented and individual characters are recognized as these merged segments. In [1], we can see some problems with segmentation. While the Japanese characters are written individually, the string written by a brush(fude) is composed of characters welded together and thus it is difficult to segment an address image. Therefore, it is not sufficient enough to recognize each character in handwritten address images. The general solution is post-processing, which takes into account the information about the language structure.

Candidate rectangle character method [2] is representative of all the post-processings, but various related techniques have been proposed. Previously in [3][4], we proposed a two-stage classification system by adapting this method. The matching score of place-name string is calculated from the characters within the lattice structure. However, in the case of just using a candidate rectangle lattice, the post-processing requires huge amounts of processing time. Therefore, we have indicated that it is crucial to check the list of place-name strings in two-stages for effective post-processing [4]. Our technique has improved the trade-off between speed and precision by rough classification and fine classification of the place-name strings.

A character recognition competition for handwritten kanji characters on postcards was held by the Institute for Posts and Telecommunications Policy (IPTP) for developing techonologies in this field [5]. All proposed methods in the competition were based on the same concept. Under this concept, our technique can be considered a good solution to handwritten Japanese address recognition. However, in practice, it is necessary to recognize the place-name string from its address image [6] (Fig.1).

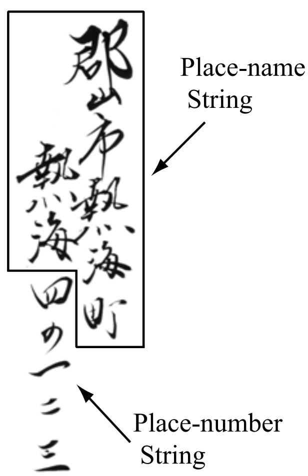

Fig. 1 Address Image

Applying the two-stage classification system to an address image, the reading accuracy goes down when the results of recognizing place-number image have influence on this system. In addition, since such system has a possibility of mismaching during the reasoning process, we consider that the recognition accuracy becomes less if the system recognizes the place-name string from its address-name image only.

In this paper, we propose the best solution to improve this problem. The main idea is to use candidate rectangle lattice and to put the results of character recognition into candidate rectangle lattice. The segments of an address image must correspond to characters within the place-name string. We design the post-processing based on phased search of candidate rectangle lattice. Taking rejection into account, the results of our experiments clearly show the effiency of our technique.

This paper is organized as follows: Section 2 describes a recognition system from an image to the string, and explains candidate rectangle lattice about our traditional mathods. The Japanese address recognition technique based on phased search of candidate rectangle lattice is proposed in Section 3, and some examples of the scores are presented. The experimental results for the IPTP 1840 address image data is reported in Section 4. In Section 5, we improves our proposed method with extended lattice search for final output. Finally some conclusions are drawn in Section 6.

2. RECOGNITION SYSTEM

Our recognition system has the following four modules.

Image Pre-processing: Pre-processing the input image (256 level gray-scale) for image segmentation into the high quality binary images. This process consists of three steps: Contrast Stretching [7], Edge Preserving Smoothing [8] and Otsu's Binarize Method [9].

Image Segmentation: Segmenting the binary image into rectangles with black dots and stroke information (cf. [1][10]).

Character Recognition: Recognizing characters with the improved directional element feature [11] and the modified maharanobis distance [12].

Post-Processing: Deducing the proper place-name string from the results of recognized characters. Fig.2 shows an outline of post-processing.

Candidate rectangle lattice

As a result of image segmentation, we call a unit segment "based rectangle" and continuous segments "merged rectangles". Both of the based rectangle and merged rectangles compose a character candidate rectangle. The address image is represented by the candidate rectangle lattice by means of a directed graph: candidate rectangle lattice is the vertex and the connection between the edge of candidate rectangles. The left part of Fig.3 shows the original image and the frame of the rectangle, and the right part of Fig.3 shows the candidate rectangle lattice.

Let $\{a, b, c, d\}$ be a set of based rectangles, $\{ab, bc, cd, abc\}$ be a set of merged rectangles, and st, en be the vertex of the start/end point of the string. On this candidate rectangle lattice, there is a plural route and each route denotes the line of candidate rectangles. In order to obtain the correct result of address recognition, a system needs to select the proper line of candidate rectangles.

Two-stage classification

Two-stage classification is the post-processing algorithm(Fig.4).

In the former steps, candidate rectangle lattice with matching scores of its characters is generated from the address image. The list of place-name strings is made beforehand from the Japanese address database. This database contains all the addresses in Japan and has about 170,000 entries, resulting in about 800,000 place-name strings. The matching score of place-name string is calculated from the characters within the lattice structure.

Let $S = s_1, s_2, \cdots, s_n$ be the place-name characters, and $X = x_1, x_2, \cdots, x_n$ be candidate rectangles. The set $U(x_i)$ consists of characters resulting by recognizing x_i $(1 \leq i \leq n)$. Inclusion in the results of character recognition is represented by $s_i \in U(x_i)$. The detection of s_i from the i-th place-name character is defined as:

$$f(s_i, U(x_i)) = \begin{cases} 1 & s_i \in U(x_i) \\ 0 & otherwise \end{cases} \tag{1}$$

The detection count is defined as:

$$\sum_{i=1}^{n} f(s_i, U(x_i)) \tag{2}$$

When the system has to narrow down the place-name lists, Eq.(3) is our standard for the elimination of unnecessary lists.

$$\sum_{i=1}^{n} f(s_i, U(x_i)) < \frac{n}{2} \tag{3}$$

It should be noted that all the parameters in this paper depend on [4].

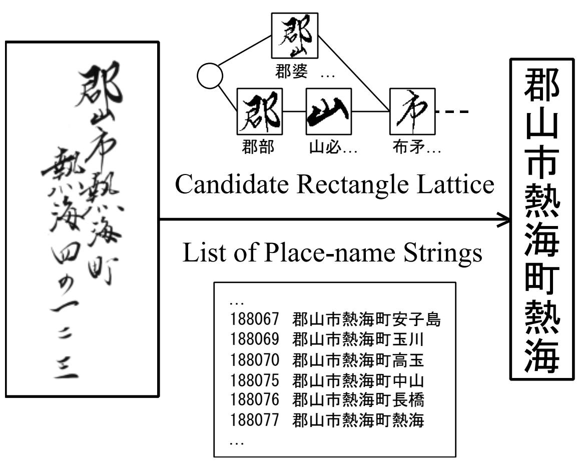

Fig. 2 Recognition System using Post-Processing

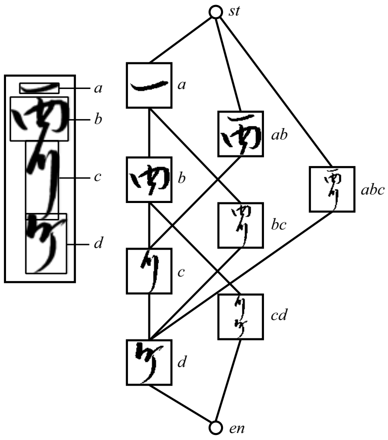

Fig. 3 Candidate Rectangle Lattice

Rough Classfication: The score of rough classfication is defined as:

$$\mathrm{L}_A(S, X) = \frac{1}{n}\left[\sum_{i=1}^{n} f(s_i, U(x_i))\right] \times 100.0 \quad (4)$$

According to Eq.(4), all the place-name lists are scoring. The system compares the score of each place-name string with the threshold. If there is only one string whose score exceeds the threshold, the string becomes the final result of the computation after rough classification. Otherwise, we proceed to fine classification of the strings with score exceeding the threshold.

Fine Classcification: During the rough classification, candidate rectangle x_i is given. $d_k(x_i)$ is the k-th distance of x_i and the ratio of $d_k(x_i)$ to $d_k(x_k)$ is denoted as the state of confidence. The score of Fine classfication is defined as:

$$\mathrm{L}_B(S, X) = \frac{1}{n}\left[\sum_{i=1}^{n} \frac{d_1(x_i)}{d_k(x_i)}\right] \times 100.0 \quad (5)$$

Final output is the string that has the best score after applying Eq.(5).

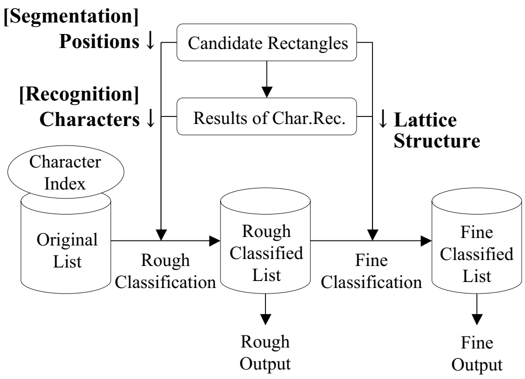

Fig. 4 Two-stage Classification

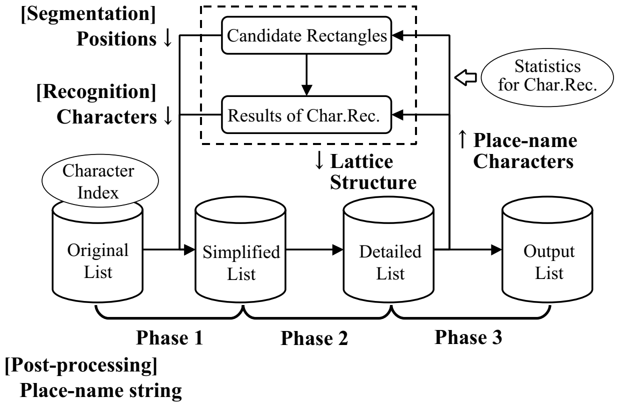

Fig. 5 Proposed Method

3. PROPOSED METHOD

In the two-stage classification [3][4], it is only sufficient to recognize the place-name string from its place-name image. This works on the assumption that the number of candidate rectangles equals to that of place-name characters. However, in this paper, that restriction must be removed. Then, it is the fact that two-stage classification doesn't keep the proper score, which is to be the best score of all place-name lists. We consider the cause to be a rectangle corresponding to the character on the place-number string. In order to recognize the characters on place-name strings with high accuracy, we improve the following two points:

- Without rough classification, searching the proper place-name on lattice to clear these correspondences.
- Re-searching lattice with recognizing the characters on the established place-name string over again.

We propose a new method based on the phased search of candidate rectangle lattice.

Outline

This method consists of the following three phases (Fig.5).

Phase 1: Referring to the character-index and flagging characters in the place-name string.

Phase 2: Searching candidate rectangle lattice with the results of character recognition.

Phase 3: Re-searching the lattice with recognizing mismatching characters.

During three phases above, the correspondence of candidate rectangles with place-name characters is confirmed.

Reducing the number of place-name lists

The detection counts narrow down the place-name lists, so we have the following two conditions.

- Correspondence with place-name string.

$$\sum_{i=1}^{n} f(s_i, U(x_i)) < \frac{n}{2}$$

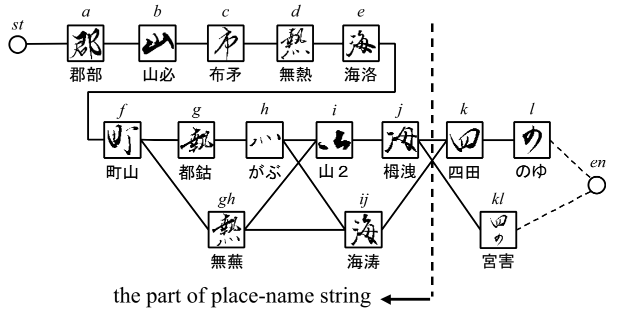

Fig. 6 Recognition Results Case 1

- Selection from candidates.

$$\sum_{i=1}^{n} f(s_i, U(x_i)) < \frac{f_{max}}{2}$$

where f_{max} is the maximum count in all the numbers of place-name lists.

Searching the Lattice

Matching the vertexes with the results of character recognition on the lattice, the candidates from place-name strings are selected. For example, we search the following two place-names from the candidate rectangle lattice(Fig.6),

- Place-name 1: $S_1 =$ 郡, 山, 市, 熱, 海, 町, 熱, 海

$$\text{海} \in U(gh) \quad \text{leads to} \quad X = a, b, c, d, e, f, gh, ij$$

- Place-name 2: $S_2 =$ 郡, 山, 市, 熱, 海, 町, 安, 子, 島

$$\begin{aligned} & X = a, b, c, d, e, f, g, h, i \\ or\ & X = a, b, c, d, e, f, g, h, ij \\ or\ & X = a, b, c, d, e, f, gh, i, j \\ or\ & X = a, b, c, d, e, f, gh, ij, k \end{aligned}$$

The rectangle with the first result of character recognition is fixed at the first search, the flagged vertex is fixed on lattice. The rectangle c on Fig.6 doesn't have the first result “市” of character recognition, but it becomes fixed automatically after the unique pass fixes the front and rear of the rectangle. Therefore, all rectangles correspond to each character of place-name string at Place-name 1. However, at Place-name 2, some candidate strings are assumed, since the first result “安, 子, 島” does not exist.

Taking a notice of this fact, we propose a new idea of phased search with correspondence between the candidate rectangle and the character of place-name string. At Phase 2, candidates are selected using only the first result of character recognition. At Phase 3, our system re-searches the lattice with recognizing mismatching rectangles in order to clear the correspondence. Furthermore, the detection counts narrow down the place-name lists, so we have the following two conditions at Phase 2.

- Correspondence with place-name string.

$$\sum_{i=1}^{n} f(s_i, U(x_i)) \leq \frac{n}{2}$$

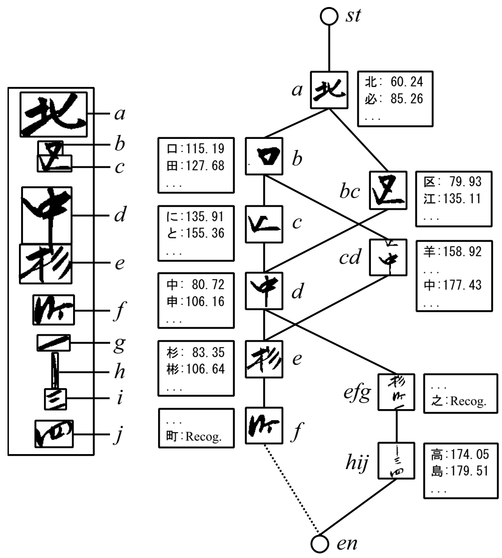

Fig. 7 Recognition Results Case 2

- Selection from candidates.

$$\sum_{i=1}^{n} f(s_i, U(x_i)) \leq \frac{f_{max}}{2}$$

Matching score

The matching score is very important in deciding the correct candidates. We mainly improve the point that the distance of character recognition has an influence on searching the lattice. That is, $\mathrm{L}_C(S, X)$ is defined as:

$$\mathrm{L}_C(S, X) = \frac{1}{f_{max}} \left[\sum_{i=1}^{n} f(s_i, U(x_i)) \right] \times \frac{1}{n} \left[\sum_{i=1}^{n} \frac{d_1(x_i)}{d_k(x_i)} \right] \times 100.0 \quad (6)$$

Since we have introduced f_{max} into searching the lattice, Eq.(6) represents the detection rate of the characters.

Example

This section gives examples of values $\mathrm{L}_A, \mathrm{L}_B$ and L_C, which are defined in sec.2 and sec.3. Fig.7 is the candidate rectangle lattice X. There are three place-names below.

- Correct place-name string:
  $S_0 =$ 北, 区, 中, 杉, 町
  (This is just, as a result of recognition)

- Incorrect place-name string 1:
  $S_1 =$ 北, 区, 中, 之, 島
  (This is long, as a result of recognition)

- Incorrect place-name string 2:
  $S_2 =$ 北, 田, 中
  (This is short, as a result of recognition)

Applying Eq.(4)(5) to these three strings, we are able to get the following values.

Rough classification[4]

- $\mathrm{L}_A(S_0, X) = 80.00$
- $\mathrm{L}_A(S_1, X) = 80.00$
- $\mathrm{L}_A(S_2, X) = 100.0$

$$\therefore \quad \mathrm{L}_A(S_2, X) > \mathrm{L}_A(S_1, X) = \mathrm{L}_A(S_0, X)$$

Fine classification[3]

- $\mathrm{L}_B(S_0, X) = 60.14$
- $\mathrm{L}_B(S_1, X) = 59.64$
- $\mathrm{L}_B(S_2, X) = 83.48$

$$\therefore \quad \mathrm{L}_B(S_2, X) > \mathrm{L}_B(S_0, X) > \mathrm{L}_B(S_1, X)$$

Both outputs of using L_A and L_B are S_2. But these results are mismatching. Because these equations have incorrect long or short place-name strings, they give an advantage over correct just place-name strings. Although L_A can't distinguish S_0 and S_1, L_B can distinguish S_0 and S_1. Therefore, according to these values, the scores of character recognition would improve the result of classification.

Applying Eq.(6) to those place-name strings, the output is S_0. By using $f_{max}(=5)$, Eq.(6) leads to the correct result.

Proposed method

- $\mathrm{L}_C(S_0, X) = 95.02$
- $\mathrm{L}_C(S_1, X) = 86.92$
- $\mathrm{L}_C(S_2, X) = 69.95$

$$\therefore \quad \mathrm{L}_C(S_0, X) > \mathrm{L}_C(S_1, X) > \mathrm{L}_C(S_2, X)$$

4. EXPERIMENTS ON RECOGNITION SYSTEM

Applying our proposal to the IPTP 1840 image data of address strings [5], the result of experiments is expressed in the Error-Reject Curve and Point "S". S is a system cost, defined as:

$$\mathrm{S} = 10\mathrm{E} + \mathrm{R}$$

where E denotes the error rate ($\%$) and R denotes the reject rate ($\%$). We set two thresholds: THD and THR. THD means the threshold for the value of the maching scores. THR means the threshold for the ratio of distances between the maching scores. These thresholds can be written as the following conditions.

- $\mathrm{L}(S_1, X) < \mathrm{THD}$
- $\mathrm{L}(S_1, X) < \mathrm{L}(S_2, X) \times \mathrm{THR}$

where $\mathrm{L}(S_1, X)$ is the best score, $\mathrm{L}(S_2, X)$ is the second best score. We can plot the experimental results obtained by assigning different values to thresholds THD and THR. On the graph in Fig.8, we only show the optimal curve. We define optimal as the monotone decreasing curve that has the lowest R when E is constant.

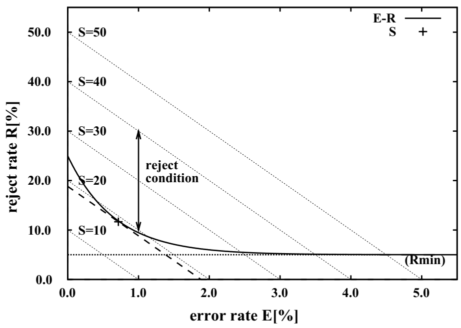

Fig. 8 E–R Curve

Experimental Results

In the old system using L_A and L_B, S is 78.75. In the new system using L_C, S is 29.78. Therefore the improvement is $\Delta\mathrm{S} = 48.97$. When we investigate the mismatching data by using L_A and L_B, the number of the strings that have an error is 76. The errors in 53 strings are mainly caused by estimating the short string. Specifically, 51 strings become correct strings by using L_C.

Fig.9 clearly shows the efficiency of our system. It should be pointed out that the highest value of S was about 60 in the IPTP competition.

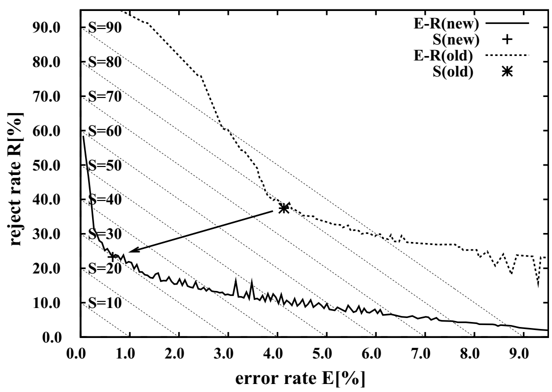

Fig. 9 Experimental results for the IPTP 1840 data

5. IMPROVED METHOD

For the strings that were mismatched, we carried out a study for the reason behind the errors. We found out that second order ones were true for most of the cases and that the proposed method needed further modification. We will discuss about this in this section.

Extended Lattice Search

In the address image shown in Fig.1, there is a part called place-number string. This part is not used in common recognition system. But, for further improvement, we will implement lattice search for this part too.

We call this lattice 'Extended Lattice' (Fig.10). In our priliminary study [13], we have found that, the 30 key-characters shown in Table 1 are included in extended lattice in most of the cases.

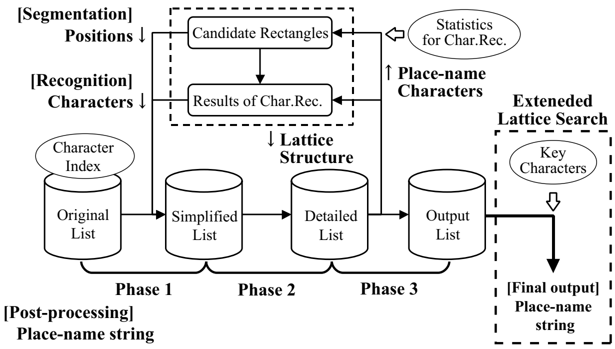

Fig. 10 Improved Method

Table 1 the 30 key-characters in the place-number string part

Numeral characters	0 1 2 3 4 5 6 7 8 9
Chinese characters	〇一二三四五六七八九十
Special characters	丁目番地号の｜〜ノ

In Output List, we consider the score of first order to be L_{C1} and the score of second order to be L_{C2}. When the ratio ($\mathrm{L}_{C1}/\mathrm{L}_{C2}$) of these scores is less than r_i ($r_i = 1.2$), we decide it as resemblance and when these key-characters are detected, further calculation is done. Next, the new value of scores are used for final output.

Example about improved method

Fig.11 is the candidate rectangle lattice X. There are two place-names below.

- First order place-name string:
  $S_4 =$ 大, 野, 町, 大, 手
  (This is incorrect. The length of S_4 is $5(= n_4)$.)
- Second order place-name string:
  $S_5 =$ 大, 野, 町
  (This is correct.The length of S_4 is $3(= n_5)$.)

We calculate the scores of S_4 and S_5 using the method mentioned in Section 3, as follows.

Proposed method

- $\mathrm{L}_C(S_4, X) = 92.90$
- $\mathrm{L}_C(S_5, X) = 88.77$

$$\therefore \quad \mathrm{L}_C(S_4, X) > \mathrm{L}_C(S_5, X)$$

Here, the ratio is 1.05 which is less than r_i. Therefore, as shown in Fig. 11, $\Delta n = |n4 - n5| = 2$ and,

$$1 \in U(e), 六 \in U(f)$$

So, we recalculate the scores, as follows.

Improved method

- $\mathrm{L}'_C(S_4, X) = 92.90$
- $\mathrm{L}'_C(S_5, X) = 98.77$

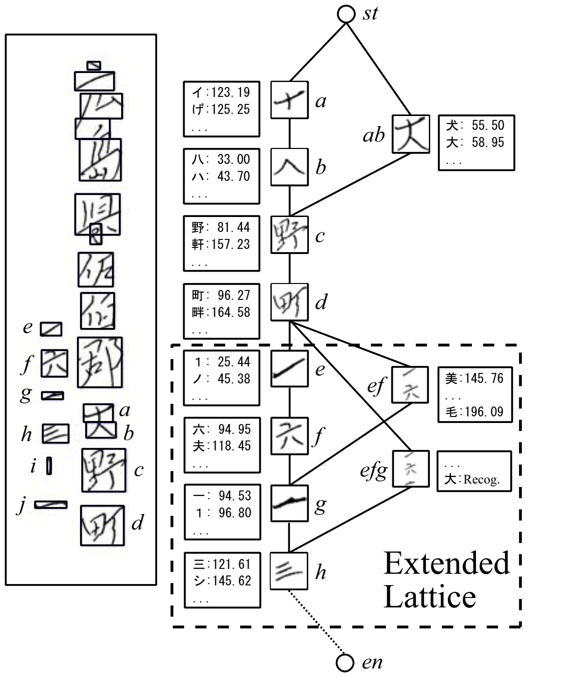

Fig. 11 Recognition Results Case 3

$$\therefore \quad \mathrm{L}'_C(S_4, X) < \mathrm{L}'_C(S_5, X)$$

In this way, for the strings having resemblance, by solving the problem that arises when $\Delta n > 0$, improvement of the results of Final output is possible.

Experimental Results about improved method

Applying our improved proposal to the IPTP 1840 image data of address strings. The results are gathered in Fig.12, which shows the percentage of correct strings without rejection.

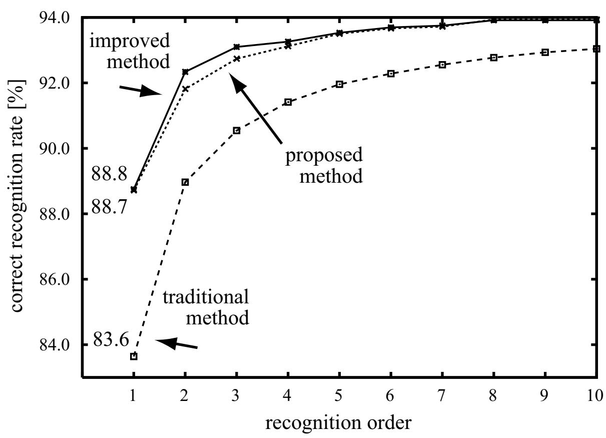

Fig. 12 Correct recognition rate (accumulated recognition rate)

Apparently, the proposed method is better compared to the traditional method. As a whole, in case of the improved method, even better results compared to those of the proposed method were obtained. For example, an improvedment of 0.5 point (10 strings) was possible for Second order. The difference between the proposed method and the

Improved method may look nominal, but because the number of the entire postal cards in Japan is well over 75 hundred million (2003, [6]), the 0.1% equivalent is 7.5 million. Thus, from the viewpoint of the actual postal environment, this difference can not be neglected at all.

6. CONCLUSION

In this paper, we applied the two-stage recognition approach to address image recognition and discussed related issues. The idea behind our proposal is to use a candidate rectangle lattice with recognizing the characters that could not be successfully recognized at the first phase of recognition. The obtained results are used to estimate the value of place strings. By so doing, we could remarkably improve the address recognition accuracy. We implemented our proposal on a handwritten address recognition system, and the experimental results display the terrific performance of our proposed method in terms of system cost (S $= 29.78$). In addition, we improved our proposed method using extended lattice search with the detection of the key-characters.

We proposed the high accuracy handwritten Japanese address recognition system. We would like to emphasize that in our studies, we mainly focused on the recognition accuracy and that more work should be done on the response time of the system. Because non-character rectangles often result in mismatching in our system, we think that the future task in this field is further research in Image segmentation. Furthermore, the paper [14] has used 3589 samples for the experiment. So, we should also consider about using larger datasets.

Acknowledgement

We would like to acknowledge the Institute for Posts and Telecommunications Policy for providing the IPTP 1840 address image data and the Advanced Industrial Science and Technology for providing the ETL character database.

References

[1] H. Ino, K. Saruta et al.: ”Handwritten address segmentation algorithm based on stroke information”, **Trans. IPS Japan**, Vol. 38, No. 2, pp. 280–289 (1997–2).

[2] H. Murase, M. Shinya et al.: ”Segmentation and recognition of hand-written character string using linguistic information”, **Trans. IEICE Japan**, Vol. J69-D, No. 9, pp. 1292–1301 (1986–9).

[3] M. Suzuki, H. Abe et al.: ”Two-stage method of handwritten address recognition using dynamic abstraction of character candidates”, **Trans. IEICE Japan**, Vol. J82-D-II, No. 11, pp. 1895–1902 (1999–11).

[4] K. Tokumoto, M. Suzuki et al.: ”Recognition system for handwritten address using rough classification method accelerated by applying priority to address candidates”, **Trans. IEICE Japan**, Vol. J84-DII, No. 1, pp. 83–92 (2001–1).

[5] F. Kawamata, T. Wakahara et al.: ”The results of the third IPTP character recognition competition for handwritten kanji character on post cards”, **Proc. of IEICE Fall Conf. ’94**, D-321 (1994–9).

[6] Ministry of Posts and Telecommunications: **Newly established Japanese postal codes manual**.

[7] M. Takagi and H. Shimoda(supervised): **”Image analysis handbook”**, University of Tokyo Press (1991).

[8] M.Nagao and T.Matsuyama: ”Edge preserving smoothing”, **Computer Graphics and Image Processing**, Vol. 9, pp. 374–407 (1979–4).

[9] N. Otsu: ”An automatic threshold selection method based on discriminant and least squares criteria”, **Trans. IEICE Japan**, Vol. J63-D, No. 4, pp. 349–356 (1980–4).

[10] H. Nakayama, M. Suzuki et al.: ”A study of effective character segmentation for handwritten address recognition”, **2001 Tohoku-section joint convention of institutes of electrical and information engineers, Japan**, 2G-17 (2001–8).

[11] N. Sun, M. Abe and Y. Nemoto: ”A handwritten character recognition system by using improved directional element feature and subspace method, **Trans. IEICE Japan**, Vol. J78-D-II, No. 6, pp. 922–930 (1995–6).

[12] N. Kato, M. Abe and Y. Nemoto: ”A handwritten character recognition system using modified mahalanobis distance”, **Trans. IEICE Japan**, Vol. J72-D-II, No. 1, pp. 45–52 (1996–1).

[13] H. Nakayama, M. Suzuki et al.: ”An inference method of place-name part for handwritten address recognition”, **2000 Tohoku-section joint convention of institutes of electrical and information engineers, Japan**, 2D-6 (2000–8).

[14] C.-L. Liu, M. Koga et al.: ”Lexicon-driven segmentation and recognition of handwritten character strings for Japanese address reading”, **IEEE Trans. on Pattern Analysis and Machine Intelligence**, Vol. 24, No. 11, pp. 1425–1437 (2002–11).