International Symposium on Communication and Information Technology (ISCIT2008) Vientiane, Lao PDR. pp.473-476

On-line Handwritten Lao Character Recognition by using Dynamic Programming Matching

Latsamy Saysourinhong[I], Bilan Zhu[II] and Masaki Nagakawa[III]
Tokyo University of Agriculture and Technology
Koganei-shi, Tokyo, 184-8588 Japan
Tel: +81-423-887144, Fax: +81-423-887144
E-mail: [I]saysourinhong@gmail.com, [II]zhubilan@cc.tuat.ac.jp, [III]nakagawa@cc.tuat.ac.jp

***Abstract*— This paper describes a method for on-line handwritten Lao character recognition by using Dynamic Programming Matching (DPM). It extracts feature points along the way from pen-down to pen-up, then uses DPM to match those feature points with the feature points for a template pattern of each character class and obtains a similarity for each character class. It selects the character class with the largest similarity as the recognition result. We also compare the recognition method by using DPM with that by the Linear-Time elastic Matching (LTM). An evaluation on the Lao character pattern database shows the result that the speed gain by LTM is small and DPM brings a better and more accuracy recognition rate.**

I. Introduction

For the past several years, the research in the fields of on-line handwritten character recognition has become a hot topic for pattern recognition. By adopting suitable techniques, better and higher accuracy recognition rates have been obtained for Latin-scripts, Japanese characters, and Chinese characters. For Indian scripts, Prasanth et al. adopted elastic matching using local feature, and obtained promising accuracy [1]. For Lao characters, Khampheth et al. reported a promising recognition rate by segmenting the input character pattern according to the changed in direction of curve clockwise or counter clockwise [2]. However, it only applied the method to 27 Lao consonants in the experiment.

In this paper, we examine established methods for the Japanese characters to the Lao characters (including consonants, vowels and tonal marks).

In this paper, section 2 presents the brief information of the Lao characters and a sample pattern collecting tool. Section 3 describes normalization and feature point extraction. Section 4 describes the recognition algorithm. Section 5 presents experimental results. Section 6 concludes the result of this paper.

II. Lao Character and Sample Pattern Collection

A. Lao Script Set

In the Lao language, the word contains syllables composed of vowel, consonant, and tonal mark (A tonal mark is used to characterize the change of syllable sound to short, medium, low or high tone.) The consonant consists of 27 single and 6 mixed consonants as shown in Fig. 1. Table 1, shows the vowels along with tonal marks and the place where it can appear in when writing the Lao language.

Fig. 1. Consonants in printed style.

TABLE I

VOWELS AND TONAL MARKS

Symbol	Type	Appear in
ຳ ະ າ	Vowel	place after consonant
ິ ີ ຶ ັ ໍ ື ົ	Vowel	place above consonant
ຸ ູ	Vowel	place below consonant
ໄ ໃ ໂ ເ ແ	Vowel	place before consonant
່ ້ ໊ ໋	Tonal mark	place above consonant

The combination of vowel itself will produce a new vowel and sometimes we also have a combination of vowel and consonant to create another vowel sound. When writing Lao language, the vowel can be placed either in front, above, right side, and below of the consonant in the word.

B. On-line Pattern Collection Tool

In this research, we focus on recognizing the individual Lao character (including 52 character classes, consonants, vowels and tone marks). For the experiment, we have prepared a tool to sample on-line handwritten Lao character patterns. Then, we created a database of handwritten Lao character patterns from 49 native Lao language speakers and 1 from a non-native speaker. Each of them wrote each character for 10 times. Fig. 2, shows printed Lao characters and corresponding handwritten patterns. All the writers were instructed to write each character within a given writing box. They was also encouraged to write in natural handwriting style.

Fig. 2. Lao character patterns sample and the data collecting tool.

III. NORMALIZATION AND FEATURE POINTS EXTRACTION

We normalize an input pattern linearly by converting the pen trace pattern to a standard size, preserving the horizontal-vertical ratio.

After the normalization, we extract feature points using the method by Ramner [3]. Ishigaki et al. employed a similar method [4, 5]. First, the starting point and the end of every stroke are picked up as the feature points. Then, the most distance point from the straight line between adjacent feature points is selected as a feature point if the distance to the straight line is greater than a threshold value. This is applied recursively until no more feature point is selected. The feature points extracting process is shown in Fig. 3.

IV. RECOGNITION ALGORITHMS

Two matching algorithms: DPM and LTM are used.

A. DPM

The matching algorithm works on the extracted feature points. The algorithm searches for correspondence between the template and input patterns.

Conventionally, elastic matching has been realized by the dynamic programming. Its search space is described as shown in Fig. 4. Roughly speaking, DPM starts from the bottom-left corner and goes right, up, or top-right directions toward the top-right corner. The matching score E(i, j) is evaluated for each pair of matching feature points as follows:

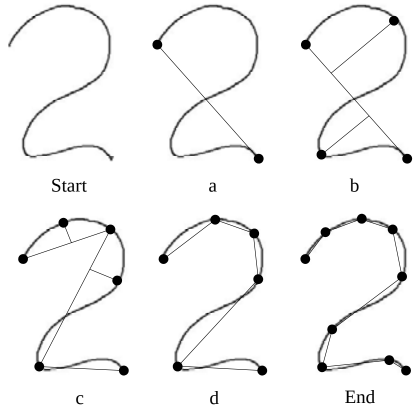

Fig. 3. Feature points extraction.

$$E(i,j) = \omega e_p(i,j) + (1-\omega) e_d(i,j) \tag{1}$$

Where:

$e_p(i,j)$: An evaluation value of the distance between template pattern's i-th point and the input pattern's j-th point. The closer the positions, the larger value assigned.

$e_d(i,j)$: An evaluation value of the angular different between the vector from the template's (i-1)-th point to i-th point, and that from the input's (j-1)-th to j-th point. The closer the angulars, the larger value assigned.

ω : Weighting parameter.

The weighting parameter is optimized by using training patterns. The path with the maximum matching scores E(i, j) for all correspondences is selected. It still requires O(mn) time complexity, where n and m are the lengths of matched feature vectors.

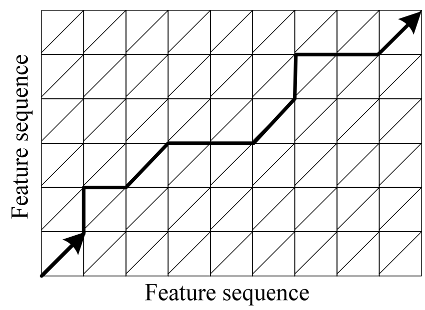

Fig. 4. The search space by DP-matching.

B. LTM

LTM was designed to speed up the recognition time for Japanese character sets of more than several thousand classes. The matching process will be as follows:

(1) The starting feature points of the input and a template are paired.
(2) [n, m] is defined as the paring of the template pattern's n-th point and the input's m-th point. Where the previous pair is [I, J] the next pair, [i, j], is decided by the following rules:

For [i, j], choose the possible pair from [I+1, J+1], [I, J+1], and [I+1, J]. Where, E(i, j) as shown in Eq. (1) is the maximum value.

(3) Matching ends when the ends of both the template and input has been reached.

With the LTM algorithm as described above, there often occurs mismatching where patterns have unstable directional features. Directional features are stable in most cases and therefore vital information for character recognition, but this information is useless in case that directional feature is unstable.

The problem can be solved by a kind of very limited and shallow backtracking (remake of feature point correspondence) as follows:

When pairing feature points, the following process is carried out if one-to-many feature point correspondence is made. The pattern containing the single point matched to many is referred to as A, while the pattern containing the matched multiple points is referred as B as shown in Fig. 5.

(1) In pattern A, let the distance between the matched single point and its previous point be L_A.
(2) In pattern B, let the sum of the distance between the multiple points and their previous point be L_B.

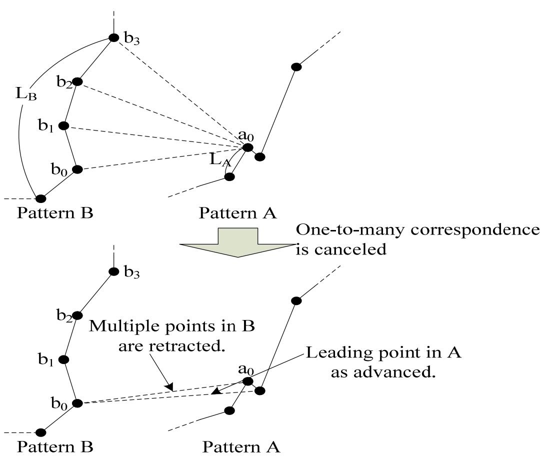

Fig. 5. An example of shallow backtracking.

(3) If (L_B-L_A) exceeds a threshold value L_S, the one-to-many correspondence is canceled. That is, the matching is retracted or backtracked and the first point of the multiple points in pattern B is forced to be paired with the next point of the single matched point in pattern A.

Only if the specify conditions occur, correspondence is retracted to a very shallow level. Once one-to-one feature point correspondences are fixed, it is never canceled and retraction does not go back beyond those correspondences. Moreover, even if retraction is made, new correspondence is proceeded which will not be canceled, so that the algorithm take a linear time complexity.

C. *Calculation of Similarity*

After the feature point correspondence is made, the algorithm calculates the similarity between the input pattern and the template pattern. The template pattern with the largest similarity is chosen as the recognized result.

We examined two methods to calculate the similarity. A method is according to one-to-many correspondences while the other is according to one-to-one correspondences.

1. Method according to one-to-many correspondences

As the result of DMP or LTM, one-to-many correspondence between feature points has been made. The algorithm calculates the average of matching scores E(i, j) for all correspondences as the similarity between input and template patterns.

2. Method according to one-to-one correspondences

Before calculating the similarity between the input and the template pattern, inappropriate correspondences are discarded and one-to-one correspondence are made. The following process is applied:

(1) Character start to character start and character end to character end correspondences are fixed.
(2) Stroke start to stroke end correspondences are discarded..
(3) Stroke end and stroke end correspondences are fixed.
(4) If there are unfixed correspondence to a feature point which has a fixed correspondence, those unfixed will be discarded.
(5) One-to-one correspondence is fixed.
(6) An unfixed correspondence is searched from character end to start and fixed. If there are unfixed correspondence to the feature points of the unfixed correspondence, those extra correspondence are discarded. This process is repeated until there is not unfixed correspondence.

An example is shown in Fig. 6.

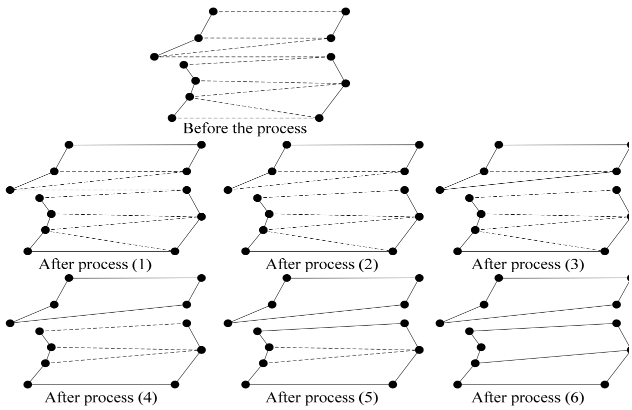

Fig. 6. One-to-one matching process.

After one-to-one correspondence is made, the algorithm calculates local degree of similarity and takes their weighted sum. This weighted sum becomes the similarity between input and template patterns. The local similarity comes from the product of an evaluation of positional and directional similarity.

V. EXPERIMENTAL RESULTS

For our experiment, handwritten patterns were collected from 49 Laotian native speakers and Japanese native speaker, where each has written 10 times per character. As a result, we have collected sample of 15,000 consonants, 9,000 vowels, and 2,000 tonal marks in our database.

In the experiment, the sample patterns from 20 people were used as training set (SET1) while the rest of patterns from 30 people were used as a test set (SET2). We obtained the recognition dictionary by recognizing each pattern of SET1 and adding the misrecognized patterns to the dictionary as template patterns. Then we also trained the weighting parameter in Eq. (1) by using SET1. Lastly, we applied the recognizer to SET2 to evaluate the performance. The results are shown in Table 2.

TABLE II

THE EXPERIMENT RESULTS

Matching Method		Data Set	Recognition rate (%)	Recognition time (s)
One-to-One	LTM	SET2	65.07	0.218
	DPM	SET2	57.32	0.257
One-to-Many	LTM	SET2	76.98	0.115
	DPM	SET2	78.06	0.154

As we can see from the experimental results, for both DPM and LTM, the similarity evaluation method according to one-to-many correspondences brought better recognition performance remarkably than that according to one-to-one correspondences. We think that this is because the processes of one-to-one correspondences destroyed the optimized correspondences between the feature points. For the better similarity evaluation method according to one-to-many correspondences, DPM brought about better recognition rate, although its processing time is behind LTM. However, the gap of the processing time between DPM and LTM is not that much, because LTM was designed to speed up the recognition time for Japanese character sets. Each stroke in the Lao character patterns is much shorter than those in the Japanese character patterns, thus the speed gain by LTM was not exploited.

VI. CONCLUSIONS

We described a method for on-line handwritten Lao character recognition by using Dynamic Programming matching (DPM) and Liner-Time elastic Matching (LTM). Our method extracted feature points then used DPM or LTM to match those feature points with the feature points for template pattern of each character class. We compared the recognition method by using DPM with LTM. The experiments showed DPM brought a better recognition rate. We will improve the recognition accuracy by applying a HMM or MRF model.

ACKNOWLEDGMENT

This research is being partially supported by the MEXT Research and Education Fund for Promoting Research on Symbiotic Information Technology.

REFERENCES

[1] L. Prasanth, V. Jagadeesh Babu, R. Raghunath Sharma and G.V. Pradhakara Rao "Elastic Matching of Online Handwritten Tamil and Telugu Scripts Using Local Feature," *Proc. of ICDAR'07*, vol. II, pp. 1028-1032, September 2007.

[2] B. Khampheth, B. Kruatrachue and S. Wangsiripitaka, "On-line Lao Handwritten Recognition with Proportional Invariant Feature," *World Enformatika Society. Transaction on Engineering, Computing and Technology,* vol. 5, pp. 41-44., April 2005.

[3] U. Ramer "An Iterative Procedure for the Polygonal Approximation of Plan Closed Curves" *Computer Graphics and Image Processing,* vol.1, pp244-256, 1972.

[4] K. Ishigaki and T. Morishita "A Stroke Number Free Character Recognition using Indispensable Feature Point," *The 33rd Annual Convention IPS Japan.*, pp. 1633-1634, October 1986.

[5] K. Ishigaki and T. Morishita "A Top-Down Online Handwritten Character Recognition Method via the Denotation of Variation," *Proc. Of Int'l Conf. on Computer Processing on Chinese and Oriental Languages*, pp. 141-145, August 1988.