Proceedings of International Conference on Image
Processing and Character Recognition

A Perturbation Technique for Solving the Handwriting Variation Problem in On-Line Handwritten Chinese Character Preclassification

Zen Chen, Chi-Wei Lee, and Rei-Heng Cheng

Institute of Computer Science and Information Engineering,
National Chiao Tung University, Hsinchu, Taiwan 30050

Abstract

In this paper we propose a method based on the perturbation technique to handle the variations in handwritten Chinese characters. The basic concept of the perturbation technique is to recover the possible erroneous stroke features (including stroke types, stroke spatial relation, stroke number, and stroke merging) with a hope that the resultant character features can find a correct match in the database. The perturbation technique can also be applied to the reference samples in the database to enhance the representation power of the database. Experiments have been conducted on a test set containing 914 different characters. Each character is written by 61 different persons with no rigid writing constraints. The correct preclassification rate is 81.42% without perturbation and 96.49% with perturbations applied to both the input samples and the database.

Keywords: Chinese character preclassification, Handwriting variations, Perturbation technique.

1. Introduction

There are two major categories of character recognition methods: the statistical approaches and the structural approaches. The advantage of the statistical approaches is the features are easy to extract, such as cellular features [2,3], projections [4,5], and two-dimensional transformations [6,7]. The disadvantage of this type of approaches is its limited recognition ability in handling similar and shape-deformed characters. On the other hand, the structural approaches [8-10] can generally yield much better recognition rate for handwritten characters. The disadvantage of this type of approach is that the features used are difficult to extract and easy to vary in handwritten characters.

A major factor for successful handwritten Chinese character recognition is the ability to handle the variations of handwriting among different writers. There are the variation of stroke types, the variation of stroke spatial relations (intersections, L-connections, T-connections, nonconnective spatial relations such as top-bottom and left-right relations), the variation of stroke number, the stroke touching or merging, and the stroke breaking, etc.

The common methods used to resolve the handwriting variations include:

(1) Imposing writing guideline [11],
(2) Using similarity matching [12-13],
(3) Using tree search [14-15],
(4) Guessing possible variations [16-17].

In this paper we propose a method based on the perturbation technique to handle the variations in handwritten Chinese characters. The basic concept of our perturbation technique is to recover the possible erroneous stroke features (including stroke types, stroke spatial relations, stroke number, and merging strokes) with a hope that the substituted character features can find a correct match.

2. Design of the Perturbation Technique

The stroke sequence of a Chinese character is very useful in Chinese character recognition. The stroke sequence provides discriminative information to classify characters. Besides, the one dimensional string representation is also suitable for string search. The preclassification method that is based on a previous method developed by us [1] is briefly described below.

First, the strokes in the character samples are extracted. There are twelve different types of curved and linear strokes. Fig. 1 shows the defined stroke types and their corresponding examples.

Second, the stroke spatial relations are also determined. The stroke spatial relations include the intersection relation, the L-relation, and the positional relations (the top-bottom relation, the left-right relation, the upper-left-and-bottom-right relation, and the bottom-left-and-upper-right relation). The L-

relation is determined by checking if the ends of two strokes are close. The positional relations between two strokes are determined by checking the overlap of the projections of the two strokes on the X-axis and Y-axis. The detailed method to extract the stroke types and the stroke spatial relations are described in [1]. The stroke types and the spatial relations of the strokes in an example character sample are listed in Fig. 2.

Third, according to a set of simple stroke ordering rules derived from the prevailing conventions for Chinese handwriting, a one dimensional stroke sequence code and a stroke structural sequence code of the character sample are generated. The detailed description of the stroke ordering rules can be found in [1]. The extended BNF of the syntax of the stroke structural sequence code is listed below. (The "{}" symbol denotes a repetitive set of symbol strings.)

<sss> ::= <unit> { <relation> <unit> }
<unit> ::= <stroke> | <X-group>
<stroke> ::= H | V | L | R | G | E | Z | S | M | W | A | B
<X-group> ::= (<stroke><relation> <stroke> { <relation> <stroke> })
<relation> ::= x | l <positional-relation> | <positional-relation>
<positional-relation> ::= ↑ | ↗ | → | ↘ | ↓ | ↙ | ← | ↖

The stroke structural sequence code <sss> describes the order of structural units and the stroke spatial relation between the last stroke of the preceding structural unit and the first stroke of the succeeding structural unit. A structural unit <unit> in a character sample is either an isolated stroke or a group of intersecting strokes. If the structural unit is an isolated stroke, its stroke type is described by <stroke>. If it is an intersection group <X-group>, the stroke type of each stroke and the stroke spatial relation between consecutive strokes in the intersection group are described. The entire substring of the intersection group is delimited by a pair of parentheses. The symbol <relation> denotes the stroke spatial relation between two consecutive strokes in the stroke sequence. It may be the intersection relation ("x"), the L-connection relation ("l") followed by a positional relation or just a positional relation. Fig. 3 lists some character samples with their corresponding stroke sequence codes and stroke structural sequence codes.

Fourth, the preclassification code of a character sample is generated from the stroke structural sequence code that is in the form of the stroke number in the character sample followed by the stroke types of the curved strokes in the order as they shown in the stroke structural sequence code. Fig. 3 also lists the preclassification codes of the character samples. The preclassification codes of representative character samples can be sorted and stored in a database so that later an unknown input character sample can be preclassified by a simple binary search method.

It can be seen that the preclassification codes of different samples of a character may somewhat change due to the variations of stroke types, stroke spatial relations, stroke number, stroke merging, or stroke breaking.

The basic principle of the perturbation technique is to endow the system to have the ability to locate the possible erroneous stroke features and recover them. Therefore the stroke feature extraction process should provide multiple possible feature values instead of just one so that the feature values can be used in perturbation. Besides, the possible feature values are preferably ordered according to their likelihood. The precedence can reduce the number of trials in recovering the erroneous stroke features and thus improve the efficiency of the perturbation.

In the following we propose a feature extraction procedure for our perturbation technique for each type of variation with clues for stroke feature correction.

(1) Perturbation on stroke types:
We use two parameters, the line segment length *LEN* and the angle between two consecutive line segments *ANG*, to determine the stroke type. Based on the definition of the twelve stroke types, when different thresholds *TLEN* and *TANG* are used to specify the length and angle requirements, different stroke types are often obtained. Thus, for a given input stroke, we may give it multiple stroke type values such as {most likely, less likely, least likely} or {nominal, variant} in short.

(2) Perturbation on nonconnective stroke spatial relations:
If there exists an uncertain bottom-left-and-upper-right relation, then the bottom-left-and-upper-right relation is selected as the nominal feature value to generate the preclassification code. The top-bottom relation is considered as the other possible feature value.

(3) Perturbation on stroke number:
To handle the variation due to the missing of the short line-segment strokes, the total number of strokes is increased and decreased by one or two. On the other hand, for the variation due to stroke merging, the stroke number is adjusted by the following stroke merging handling procedure.

(4) Perturbation on merging strokes:

The step used to resolve the simple type of stroke merging is to decompose a merging stroke into two constituent strokes. Usually the simple type of stroke merging is due to the connection of a linear stroke to another stroke. The new preclassification codes after perturbation are obtained by simply decomposing the possible merging stroke into two consecutive strokes in the stroke sequence. In the current system, every curved stroke is assumed to be a possible merging stroke.

Since there may be more than one erroneous stroke feature in the character sample, the perturbation technique must generate all preclassification codes based on all the possible combinations of stroke feature values. Furthermore, the preclassification codes better be generated one by one in a proper order so that the preclassification codes with higher probabilities of likelihood are in front of those with lower probabilities. This ordering can improve the efficiency of matching.

There are basically two guidelines to arrange the order of the preclassification codes generated by the perturbation technique:

(1) The probability that the input character sample contains n erroneous stroke features generally decreases as n increases. Hence the preclassification codes generated from the perturbation of fewer stroke features should precede those generated with more perturbed stroke features.

(2) Assume there is a total of N stroke features and n of them are selected for perturbation, there are $C\binom{N}{n}$ selections. Since the range of variation in the mild variations (stroke types and stroke spatial relations) is generally smaller than that in the wild variations (stroke number and merging strokes), the perturbation of the stroke features with mild variations has a higher probability of being restored. Therefore, the preclassification codes containing perturbed stroke features with mild variations are ordered ahead of those containing perturbed stroke features with wild variations.

Algorithm: **Preclassification codes generation by the perturbation technique**

Input:

(1) *MAX*: the maximum number of stroke features considered for perturbation.

(2) *Stype*: the set of strokes that have more than one possible stroke type in the given character sample.

(3) *Srel*: the set of stroke spatial relations that are the confusing bottom-left-and-upper-right relations in the given character sample.

(4) *Smerge*: the set of curved strokes in the given character sample.

(5) *Fstroke_num*: the total number of strokes in the given character sample.

Output:

TOTAL_CODES: the possible preclassification codes ordered in the descending order of the probability of likelihood.

Method:

1. Let *CODES(0)* contains the preclassification code generated without perturbation.
2. For $n = 1$ to *MAX* do
2.1. Let *CODES(n)*, the preclassification codes generated by perturbing n stroke features, be initially an empty set.
2.2. Select n stroke features from the sets *Stype*, *Srel*, *Smerge*, and *Fstroke_num* for perturbation. There are $C\binom{|S_{type}|+|S_{rel}|+|S_{merge}|+1}{n}$ selections. Let U be the set of these selections.
2.3. Order the selections in U so that the selections containing perturbed stroke features with mild variations are ordered ahead of those with stroke features with wild variations.
2.4. For each selection u in U do
2.4.1. For each combination of the perturbed stroke feature values in u do
2.4.1.1. If a stroke feature in u is from *Stype*, perturb it to become one of its new possible stroke types.
2.4.1.2. If a stroke feature in u is from *Srel*, perturb it to become its new possible stroke spatial relation.
2.4.1.3. If a stroke feature in u is from *Smerge*, decompose it into two strokes and increase *Fstroke_num* by one.
2.4.1.4. If a stroke feature in u is *Fstroke_num*, perturb it to become *Fstroke_num*+1 and *Fstroke_num*-1.
2.4.1.5. Generate the stroke

structural sequence code with the new set of the character stroke features.

2.4.1.6. Construct the preclassification code from the newly generated stroke structural sequence code. Append it to *CODES(n)* if it does not appear in *CODES(0), CODES(1), ...,* or *CODES(n)*.

next

next

next

3. *TOTAL_CODES* = Concatenation of *CODES(0), CODES(1), ...,* and *CODES(MAX)* in order.

3. Experimental Results

To evaluate the performance of our perturbation technique for handling the ordinary human handwriting variations, we conduct experiments for the three modes of the perturbation technique. The test data set is obtained from the Telecommunication Laboratories of Ministry of Transportation and Communication of R.O.C. The data set contains 204 characters whose stroke number ranges from 5 to 10, 604 characters whose stroke number ranges from 15 to 20, and 106 characters whose stroke number ranges from 22 to 30. Each character is written by 61 different persons with no rigid writing restrictions. Totally, there are 914 different characters and 914× 61=55,754 character samples.

Five samples out of the 61 samples in each character are selected and stored in the database and the other 56 samples are used as the input samples. Since the samples selected for the database construction will affect the preclassification outcome, we use two databases in the experiments. Database A is constructed by randomly select five out of the 61 samples of each character. Database B contains five good samples of each character in the sense that these selected samples can generate as many distinct perturbed preclassification codes as possible when the perturbation technique is applied to the database. All the experiments are executed on a Pentium-75 IBM compatible PC.

Experiment 1: Perturbation applied to the input samples (mode 1)

A. The improvement on the preclassification rate:

We analyze the improvement on the preclassification rate when perturbation is applied to the input samples. The maximum number of stroke features selected for perturbation ranges from zero to five. Based on our observation, the character samples of the test data contain more variations in stroke types and stroke spatial relations and less variation in stroke merging or breaking, so our perturbation procedure only allows at most one merging stroke in the input samples. The input sample is considered correctly preclassified if one of its perturbed preclassification code finds a match in the database, otherwise, it is not correctly preclassified. The preclassification result is listed in Table 1.

As can be seen from Table 1, when there is no perturbation, the preclassification rate is 73.24% if using database A and 81.42% if using database B. So the samples used in the database can somewhat affect the preclassification outcome.

Furthermore, the application of the perturbation technique to the input samples raises the preclassification rate up to 89.70% if database A is used and up to 94.05% if database B is used. We can see from Table 1 that the reasonable number of stroke features selected for perturbation when applied to the input samples is three or four.

Experiment 2: Perturbation applied to the database (mode 2)

As can be seen from Table 2, the application of the perturbation technique to the database raises the preclassification rate from 73.24% to 78.99% if database A is used and from 81.42% to 88.86% if database B is used.

In mode 1 we allow four types of perturbations (stroke types, stroke spatial relations, stroke number and merging strokes) performed on the input samples, so the number of possible different preclassification codes may increase to a greater extent. On the other hand, in the current version of mode 2 we only allow two types of perturbations (stroke type and stroke spatial relation) performed on the reference samples in the database, so the number of possible different preclassification codes generally does not increase as many as in the former case. However, if we also allow four types of perturbations in mode 2, then it generally gives a better result. In the future we probably will not select reference samples in the database randomly. Instead, the database can be designed off-line to contain the preselected good representative samples.

The number of varying stroke features in most handwriting samples is generally small. We can see from Table 2 that the reasonable number of stroke features selected for perturbation in mode 2 is two.

Experiment 3: Perturbation applied to both the input samples and the database (mode 3)

At most five stroke feature perturbation on the input samples and at most three stroke feature perturbation in the database are performed. The database used here is database B (If database A is used, the result is similar). From Table 3, the preclassification rate without any perturbation is 81.42%. When more stroke features are used for perturbation, the preclassification rate gets higher. If we are not concerned about the overhead of the perturbation, the best preclassification rate is around 96.49% (an increase of 15.07%).

Table 3 shows that the preclassification rate is converged around 96%. Table 4 lists part of the corresponding execution time. Table 5 gives the storage space required. If we want to achieve high preclassification rate, while taking the overhead of execution time and storage space into consideration, a good choice is to perturb at most three stroke features on the input samples and at most one stroke feature in the database. In this case, the preclassification rate is still as high as 96.06%

4. Conclusions

In this paper we propose a method based on the perturbation technique to handle the variations in handwritten Chinese characters. The basic concept of our perturbation technique is to restore the possible erroneous stroke features by replacing them with new feature values with a hope that the recovered input character sample can be preclassified correctly. The perturbation technique can be used as an aid to a recognition procedure to improve its preclassification rate. In summary, the advantages of the proposed technique include:

(1) The user has more freedom in writing characters,
(2) The system can handle more handwriting variation without enlarging the database,
(3) No need to use the time-consuming similarity computation to find the most similar character,
(4) The application modes of the perturbation technique are flexible

Three different modes of the perturbation technique can be used. The system designer can choose the appropriate mode for the system.

For the preclassification using the perturbation technique, there are still 3.51% of character samples that are mis-preclassified. The mis-preclassification of character samples is mainly due to the variation of complex types of stroke merging, missing stroke, broken strokes, and multiple standard handwriting styles. In the future we shall study these more challenging problems.

References:

[1] C. W. Lee, R. H. Cheng, and Z. Chen, "Automatic handwritten Chinese character stroke ordering and its applications", submitted to Pattern Recognition.

[2] L. Wang, "Recognition of handprinted Chinese characters using outline direction and background density", Proc. Int. Conf. on Computer Processing of Chinese and Oriental Languages, pp. 39-42, 1988.

[3] F. Kimura, K. Takashina, S. Tsuruoka and Y. Miyake, "Modified quadratic discriminant functions and the application to Chinese character recognition", IEEE Trans. Pattern Analysis and Machine Intelligence PAMI-9(1), pp. 149-153, 1987.

[4] E. Yamamoto, N. Fujii, T. Fujita, C. Ito and J. Tanahashi, "Handwritten Kanji character recognition using the features extracted from multiple standpoints", Proc IEEE Conf. on Pattern Recognition and Image Processing, pp. 131-136, 1981.

[5] Y. X. Gu, Q. R. Wang and C. Y. Suen, "Application of a multilayer decision tree in computer recognition of Chinese characters", IEEE Trans. Pattern Analysis and Machine Intelligence PAMI-5(1), pp. 83-89, 1983.

[6] Y. S. Cheung and C. H. Leung, "Chain-code transform for Chinese character recognition", Proc. Int. IEEE Conf. on Cybern. and Soc., pp. 42-45, 1985.

[7] K. Maeda, Y. Kurosawa, H. Asada, K. Sakai and S. Watanabe, "Handprinted Kanji recognition by pattern matching method", Proc. Int. Conf. on Pattern Recognition, pp. 789-792, 1982.

[8] F. H. Cheng, W. H. Hsu and C. A. Chen, "Fuzzy approach to solve the recognition problem of handwritten Chinese characters", Pattern Recognition 22, pp. 133-144, 1989.

[9] B. S. Jeng and G. H. Chang, "A PC-based recognition system for printed and handwritten Chinese characters through pipelining approach", Proc. Int. Conf. on Computer Processing of Chinese and Oriental Languages, pp. 49-53, 1988.

[10] K. P. Chan and Y. S. Cheung, "Fuzzy-attribute graph with application to Chinese character recognition", IEEE Trans. Syst. Man Cybern. 22(1), pp. 153-160, 1992.

[11] K. T. Lua, "Analysis of Chinese character stroke sequences", Computer Proc. of Chinese and Oriental Languages 4 (4), pp. 375-385, 1990.

[12] C. H. Leung, Y. S. Cheung and Y. L. Wong, "A knowledge-based stroke matching method for Chinese character recognition", IEEE Trans. Syst. Man Cybern. SMC-17(6), pp. 993-1003, 1987.

[13] S. L. Chou and W. H. Tsai, "Recognizing handwritten Chinese characters by stroke-segment matching using an iteration scheme", Int. J. Pattern Recognition and Artificial Intelligence 5(1/2), pp. 175-195, 1991.

[14] C. W. Liao and J. S. Huang, "A transformation-invariant matching algorithm for handwritten Chinese character recognition", Pattern Recognition 23, pp. 1167-1188, 1990.

[15] M. Zhao, "Two-dimensional extended attribute grammar method for the recognition of handprinted Chinese characters", Pattern Recognition 23, pp. 685-695, 1990.

[16] C. K. Lin, K. C. Fan and F. T. P. Lee, "On-line recognition by deviation-expansion model and dynamic programming matching", Pattern Recognition 26(2), pp. 259-268, 1993.

[17] C. C. Hsieh and H. J. Lee, "A probabilistic stroke-based Viterbi algorithm for handwritten Chinese characters recognition", Int. J. Pattern Recognition and Artificial Intelligence 7(2), pp. 329-352, 1993.

Table 1 The improvement on the preclassification rate when perturbation applied to the input samples.

Database A

The maximum number of stroke features perturbed	0	1	2	3	4	5
Preclassification rate	73.24%	78.71%	83.33%	88.84%	89.56%	89.70%

Database B

The maximum number of stroke features perturbed	0	1	2	3	4	5
Preclassification rate	81.42%	87.64%	91.50%	93.40%	93.97%	94.05%

Table 2 The improvement on the preclassification rate when perturbation applied to the databases A and B.

Database A

The maximum number of stroke features perturbed	0	1	2	3
Preclassification rate	73.24%	78.03%	78.86%	78.99%

Database B

The maximum number of stroke features perturbed	0	1	2	3
Preclassification rate	81.42%	87.94%	88.72%	88.86%

Table 3 The improvement on the preclassification rate when perturbation applied to both the input samples and the database.

database* \ input ■	0	1	2	3	4	5
0	81.42%	87.64%	91.50%	93.40%	93.97%	94.05%
1	87.94%	92.48%	94.95%	96.06%	96.30%	96.33%
2	88.72%	92.81%	95.17%	96.26%	96.45%	96.47%
3	88.86%	92.85%	95.19%	96.29%	96.47%	96.49%

database*: the maximum number of stroke features perturbed in database B.
input■: the maximum number of stroke features perturbed on the input samples.

Table 4 The average processing time per input sample for the preclassification method with perturbation.

database* \ input ■	3	4	5
1	35.17 ms	74.22 ms	103.65 ms
2	43.94 ms	87.80 ms	118.59 ms
3	52.83 ms	102.83 ms	138.32 ms

database*: the maximum number of stroke features perturbed in database B.
input■: the maximum number of stroke features perturbed on the input samples.

Table 5 The storage space of database B required in mode 3.

The maximum number of stroke features perturbed	0	1	2	3
Total clusters	1682	4349	8594	11524

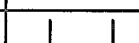
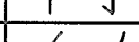
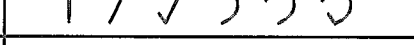
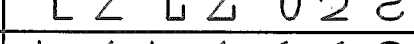
	Stroke type	Handwritten examples
Linear strokes	TYPE-H	
	TYPE-V	
	TYPE-L	
	TYPE-R	
Curved strokes	TYPE-G	
	TYPE-E	
	TYPE-Z	
	TYPE-S	
	TYPE-M	
	TYPE-W	
	TYPE-A	
	TYPE-B	

Fig. 1 Stroke types and their corresponding examples.

Stroke types include type(s1) = {L}, type(s2) = {R}, type(s3) = {G}, type(s4) = {M}, and type(s5) = {H}.
Stroke spatial relations include rel(s1,s2) = {↙}, rel(s1,s3) = { ⇕ }, rel(s1,s4) = { ⇕ }, rel(s1,s5) = { ⇕ }, rel(s2,s3) = {↔, L-connection}, rel(s2,s4) = {↘}, rel(s2,s5) = {↘}, rel(s3,s4) = {↔, L-connection}, rel(s3,s5) = { ⇕ }, and rel(s4,s5) = {intersection}.
(The symbol " ⇕ " means the top-bottom relation, "↔" means the left-right relation, "↘" means the upper-left-and-bottom-right relation, and "↙" means the bottom-left-and-upper-right relation)

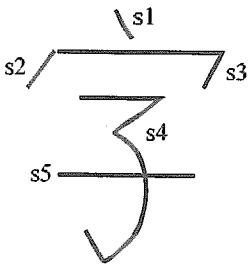
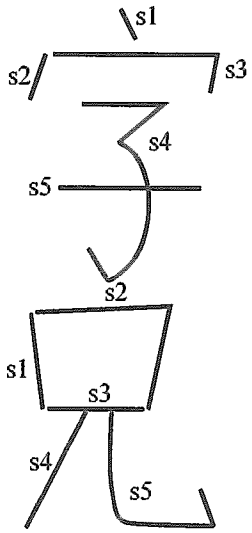


Fig. 2 Stroke types and spatial relations of an example character.



Stroke sequence code: "s2 s1 s3 s4 s5"
(whose type code is "LRGMH").
Stroke structural sequence code:
"L ↗ R ↓ G ↓ (M x H)"
Preclassification code: "5,GM"

Stroke sequence: "s1 s2 s3 s4 s5"
(whose type code is "VGHLE").
Stroke structural sequence code:
"V 1→ G 1 ↓ H ↓ L → E"
Preclassification code: "5,GE"

Fig. 3 Some character samples and their corresponding stroke codes.