TECHNIQUES FOR ON-LINE CHINESE CHARACTER RECOGNITION WITH REDUCED WRITING CONSTRAINTS

P. J. Ye, H. Hugli and F. Pellandini

Reprinted from IEEE SEVENTH INTERNATIONAL
CONFERENCE ON PATTERN RECOGNITION Proceedings
July 30-August 2, 1984 Montreal, Canada

TECHNIQUES FOR ON-LINE CHINESE CHARACTER RECOGNITION WITH REDUCED WRITING CONSTRAINTS

P.J.Ye - H.Hugli - F.Pellandini

Institut de microtechnique de l'université de Neuchâtel
71 rue de la Maladière, 2000 Neuchâtel 7 - Switzerland

1.0 ABSTRACT

Stroke-character recognition methods utilised so far for on-line handwritten chinese character recognition have many writing constraints. This paper presents the investigations made for allowing more freedom in writing by basing recognition more particularly on sampling segmentation, stroke merging and decomposing, stroke and character recognition by non-exact matching classification and stroke sequence rearrangement.

2.0 INTRODUCTION

Methods utilised so far for on-line chinese character recognition are effective for given writing rules [1],[2],[3]. Recently we investigated methods to lighten these rules [4].

Freedom in writing is provided by 1) sampling segmentation, 2) merging of strokes which were split during writing or acquisition, 3) decomposing strokes which consist of several basic strokes linked together, 4) non-exact matching at the level of both stroke and character classification, 5) rearranging the input stroke sequence and performing matching using a defined sequence.

3.0 SEGMENTATION

Segmentation follows filtering, smoothing and normalizing the stroke information. Its aim is to divide a stroke into a reasonable number of significant straight line segments. Several segmentation methods have been proposed [5]. In this paper, a successive sampling method is presented.

We define: 1) a stroke T as a sequence of points:

$$T = \underline{p}(1), \underline{p}(2), .., \underline{p}(k), .., \underline{p}(K+1)$$

2) the segment $\underline{s}(k)$ as the straight line connecting two consecutive points $\underline{p}(k)$ and $\underline{p}(k+1)$, $l(k)$ being its length; 3) the angle $a(k)$ as the angle formed by two consecutive segments $\underline{s}(k-1)$ and $\underline{s}(k)$. The basic principle of segmentation by the successive sampling method used is to transform an input stroke in an output stroke by successively discarding points from the input stroke. The sampling method includes: 1) two-threshold sampling and 2) maximum angle sampling.

Two-threshold sampling successively discards points forming small angles and short segments. The discarding decision is according to two thresholds, a_t for the angle and l_t for the segment length.

Maximum angle sampling successively discards, among all points $\underline{p}(k)$ still forming the stroke, the one point which forms the largest angle.

The method uses successively both samplings described above. First, two-threshold sampling is used to eliminate the largest part of unnecessary points. Then, maximum angle sampling is used repeatedly until the sequence is reduced to a maximum number of K= 4 segments. This is because 4 is the maximum number of segments forming the basic strokes of chinese characters. Finally, two-threshold sampling is used again to eliminate unnecessary points behond this limit, producing a final sequence of K segments, $K \leq 4$.

Practical values for the thresholds are a_t= 120 degrees and l_t= 0.1*L where: $L = l(1) + ... + l(K-1)$.

Following this sampling, a further segmentation is applied to T in order to eliminate segments which are short relative to the other segments of the stroke. Rearranging the remaining segments in order of decreasing length value, l(1), l(2), .., l(K), we define the new threshold l_r:

$$l_r = \begin{cases} h.l(1) & \text{if } K=2 \\ h.l(2) & \text{if } K>2 \end{cases}$$

practically used with a value of h= 0.33.

After segmentation, a stroke is a sequence of K distinct segments:

$$T = \underline{s}(1),....,\underline{s}(K) \qquad \text{where:} \quad K \leq 4$$

4.0 STROKE CLASSIFICATION

4.1 Direct classification

Standard chinese characters are made of strokes which are indivisible and which can be classified according to their shape in a limited number of classes. We call such strokes basic strokes B(n) and the corresponding classes, basic stroke classes we denote $b_1, b_2, .., b_i, .., b_I$. The aim of stroke classification is to classifie the input strokes into the basic stroke classes.

In the simple case, there is a one to one correspondance between the input stroke T and a basic stroke B and T can be classified directly. Practically, this is done by entering the feature vector into a classification tree [6]. If a stroke T is classified into the basic stroke class b_i, we set the class index C(n)= i.

CH2046-1/84/0000/1043$01.00©1984 IEEE

4.2 Stroke merging and decomposing

There is usually not a one to one correspondance between T and B because, with handwriting, also non-ideal input strokes must be considered, i.e. such strokes resulting from splitting a basic stroke or linking several basic strokes.

The recognition of split strokes is based on a series of tests applied to each pair of consecutive strokes T(h),T(h+1). Stroke merging is performed when both the following conditions are fulfilled: 1) the distance between the last point of T(h) and the first point of T(h+1) must be inferior to a threshold and 2) the newly linked stroke must be classifiable.

The recognition of linked strokes is made possible in a certain number of practical cases by considering additional stroke classes for linked sequences of basic strokes, i.e. compound strokes. We also associate to these classes the indices C(u(1)), C(u(2)),.. of the corresponding basic stroke classes. An input stroke T classified into such a class is then decomposed into the basic strokes accordingly.

After stroke merging, classification and decomposition, we obtain a stroke or subsequence of stokes B(n).

5.0 CHARACTER CLASSIFICATION

5.1 Principle

A character X to be recognized is given as a sequence of N classified strokes B(n), each identified by a stroke class index C(n) and characterized by a feature vector F(n):

$$X = B(1),..B(n),..,B(N) \quad \text{with: } B(n)= (C(n),F(n))$$

The recognition at character level is by reference matching, and the set of reference characters $\{R\} = \{R_m: m=1..M\}$ is obtained by training. Similarly, a single reference R_m writes:

$$R_m = B_m(1),..,B_m(n),..,B_m(N_m)$$

$$\text{with: } B_m(n)= (C_m(n),F_m(n))$$

In the following sections, two kind of alternatives are suggested:
- exact versus non-exact matching
- considering versus not considering the stroke sequence for recognition

5.2 Exact matching

Recognition is based uniquely on the stroke class sequence. Exact matching means:

$$X=R_{m*} \quad \text{if: } N=N_{m*} \text{ and } C(n)=C_{m*}(n),\ n=1..N$$

and X is classified in class m* if such a reference can be found, otherwise it is not classified.

Obviously, this approch is not successful because exact matching is a very harsh condition for handwriting and because ambiguity can result from it if two or more characters have the same stroke class sequence.

In opposition, non-exact matching performs recognition by best matching X with a reference.

5.3 Stroke features and interstroke distance

A stroke feature vector F(n) can be written:

$$F(n) = (\ Kn,\ \underline{s}(n,1)....\underline{s}(n,Kn),\ \underline{g}(n)\ )$$

it includes:
- stroke shape information given by the sequence of Kn segments: $\underline{s}(n,1)$, $\underline{s}(n,2)$, .. ,$\underline{s}(n,Kn)$ whose length l(n,k) is also used.
- stroke position information given by $\underline{g}(n)$, defined as the center of mass of the stroke.

Interstroke distances are of three kinds: shape distance d_s, position distance d_p and length distance d_l and defined next for formal strokes Bp and Bq.

Interstroke shape distance $d_s(Bp,Bq)$ combines two distances:

$$d_s(Bp,Bq) = d_k(Bp,Bq) + d_o(Bp,Bq)$$

d_k, the segment number distance is:

$$d_k(Bp,Bq) = (\ (Kp-0.5*Ap) - (Kq-0.5*Aq)\ )^2$$

where the values Ap and Aq are set to either 0 or 0.5 depending on the stoke classes C(p) respectively C(q).

$d_o(Bp,Bq)$, the segment sequence distance is the dynamic distance:

$$d_o = \min_v\{\ 1/K_{min} * \text{sum}_k\{\ d(\underline{s}(p,k),\underline{s}(q,k+v))\ \}\ \}$$

$$\text{for: } v=0..(Kp-Kq) \quad \text{and} \quad k=1..K_{min}$$

$$\text{where: } K_{min}= \min\{\ Kp,\ Kq\ \} \quad \text{and:}$$

$$d(\underline{s}(i),\underline{s}(j))= 4/pi * \text{acos}\{\ \underline{s}(i)*\underline{s}(j)/l(i)*l(j))\ \}$$

Interstroke position distance $d_p(Bp,Bq)$ is:

$$d_p(Bp,Bq) = \text{abs}\{\ \underline{g}(p)-\underline{g}(q)\ \}$$

Interstroke length distance $d_l(Bp,Bq)$ is:

$$d_l(Bp,Bq) = \begin{cases} \text{abs}\{\ l(p,1)-l(q,1)\ \} & \text{if } Kp=Kq=1 \\ 0 & \text{otherwise} \end{cases}$$

Notice that, as stated above, the interstroke length distance is non-zero only with one-segment strokes.

5.4 Intercharacter distances

Intercharacter distances lie on interstroke distances. They are of four kinds: intercharacter stroke number distance D_n, intercharacter shape distance D_s, intercharacter position distance D_p and intercharacter length distance D_l and defined:

$$D_n(X,R_m) = \text{abs}\{\ N-N_m\ \}$$

$$D_s(X,R_m) = \text{sum}_n\{\ d_s(B(n),B_m(n))\ \} \quad n=1..N$$

$$D_p(X,R_m) = \text{sum}_n\{\ d_p(B(n),B_m(n))\ \} \quad n=1..N$$

$$D_l(X,R_m) = \max_n\{\ d_l(B(n),B_m(n))\ \} \quad n=1..N$$

Notice that the three later only apply to characters with the same number of strokes ($N=N_m$). The choice of the maximum function in the definition above is motivated by the desire to consider only the most different stroke pair.

5.5 Non-exact matching classification

The classification proceeds in successive steps:

each step operates on the rest set {Rr} from the previous step, eliminates references and passes the rest set to the next step. It ends by best match classification or by rejection.

Step 1: from {Rr}={R} discard all references such that:

$$D_n(X,R_m) = 0$$

Step 2: from {Rr} discard all references such that:

$$D_s(X,R_m) > s_{th}$$

Step 3: from {Rr} discard all references such that:

$$D_p(X,R_m) > p_{th}$$

Step 4: from {Rr} discard all references such that:

$$D_t = D_s(X,R_m)/M_s + D_p(X,Rm)/M_p > t_th$$

where: $M_s = mean_m\{ D_s(X,R_m) \}$

$M_p = mean_m\{ D_p(X,R_m) \}$

Step 5: using {Rr} classifie with class m* such that:

$$m* = argmin_m\{ D_l(X,R_m) \}$$

Obviously, the three thresholds s_{th}, p_{th} and t (th) must be chosen carefully.

5.6 Without considering the stroke sequence

The order in which the strokes are written plays an important role [7], and basing the recognition on this order is usually successful. However, it supposes the user will write according to very strict rules. More freedom in writing is achieved if the recognition does not consider the order of the input stroke sequence, as will be done here.

The basic idea is to rearrange the stroke sequence according to some schema and to perform recognition on this rearranged stroke sequence. The schema presented here performs full rearrangement, i.e. the final order in independent of the order of the input stroke.

A first approach does: a) stroke sequence rearrangement according to the stroke class index; b) stroke class exact matching; c) stroke position best matching.

Step 1 is as above.

Step 2 rearranges the stroke sequence of the character X in order of increasing stroke class indices C(n). Note that the reference sequences R_m are also arranged accordingly.

Step 3 performs stroke class exact matching: it discards from the rest set {R} all references such that:

$$N=N_m \quad \text{and} \quad C(n)=C_m(n), \quad n=1..N_m$$

Step 4 is stroke position best matching: using {Rr}, X is classified in class m* such that:

$$m* = argmin_{m*}\{ D_{p*}(X,R_{m*}) \}$$

where D_{p*} is the best intercharacter position distance obtained by rearranging the stroke sequence. As the stroke sequence is already arranged in order of class indices C(n), the rearrangement here considers the permutations among strokes with the same C(n). Calling

$$Y_i = B_i(1), ..,B_i(u), ..,B_i(U_i)$$

the subsequence built with all strokes B(n) with the same class index C(n)=i, the best distance D_{p*} writes as:

$$D_{p*}(X,R_m) = sum_i\{ min_Y\{ D_p(Y(i),Y_m(i)) \} \}$$

i.e. the position distance is minimized at each subsequence level i by permutation of the strokes $B_i(u)$ belonging to subsequence Y(i).

A second, more general approach considers also mismatch at the stroke class level but cannot be decribed in detail [4].

6.0 CONCLUSIONS

The techniques and algorithms presented are implemented with an experimental character recognition system (PDP-11/60) using a tablet digitizer for on-line input of the handwritten characters. Tests were carried out on characters written by chinese persons with a reference set of 500 chinese characters (3 to 10 strokes per character) and using a small set of basic strokes. The correct recognition rate for non-exact matching is 99 % when considering the stroke sequence and 92 % when not considering the stroke sequence. Further, we estimate that the expansion of the character set up to a size of 2500 characters is straigthforward.

Especially because of stroke sequence rearrangement, stroke merging and decomposing, and non-exact matching we could experience good writing freedom.

7.0 REFERENCES

[1] Fu K.S., "Application of Pattern Handling Methods to Chinese Language Processing", School of E.E., Purdue University, USA, April 1979

[2] Nakata K. & al., "Problems in Chinese Character Recognition", Central Research Lab. Hitachi, Tokyo, Japan, First USA-Japan Computer Conference 1972

[3] Mori & al., "Advance in Recognition of Chinese Character", Toshiba Research and Development Center, Japan, (C) IEEE 1980

[4] Ye P.J., "Reconnaissance automatique des caracteres chinois manuscrits par ordinateur", Report, Institut de microtechnique de l'Universite de Neuchatel, Switzerland, December 1983

[5] Patrick & Kunt, "Efficient Coding of High Resolution Typographic Characters", EPFL Lausanne, Switzerland, (C) IEEE 1982

[6] Fu K.S., "Syntactic Pattern Recognition and Application", Prentice-Hall, 1982

[7] Shen C.Y. & al., "Automatic Recognition of Handprinted Characters - State of the Art", University Concordia, Canada, Proceedings IEEE, vol. 68, no. 4, April 1980

[8] Odaka & al., "On-line Recognition of Handwritten Character by Approximating Each Stroke with Several Points", (C) IEEE 1982