An Offline Recognition Method of Handwritten Primitive Manchu Characters Based on Strokes

Guang-yuan Zhang
School of Information Science & Engineering, Northeastern University, 110004, Shenyang, People's Republic of China;
School of Information Engineering, Shenyang University, 110044, Shenyang, People's Republic of China
E-mail: kangue@sohu.com

Jing-jiao Li
Northeastern University, 110004, Shenyang, People's Republic of China
Rong-wei He
Liaoning archives Library, 110000, Shenyang, People's Republic of China
Ai-xia Wang
Northeastern University, 110004, Shenyang, People's Republic of China

Abstract

Probing into the features of Manchu language especially its characters or letters, the stroke elements of primitive Manchu are defined. Then, a system of offline handwritten primitive Manchu character recognition method based on strokes is built. Briefly, a pre-process for primitive Manchu is operated to get single word with digital image process methods, which then decompounds the word to stroke elements, the stroke elements sequence is obtained by using statistical pattern recognition method and the Manchu-Roman code is obtained by using fuzzy string matching algorithm. Simulation results showed that the result is effective based on the features library and rules library.

1. Introduction

During the times of Nuerhachi (who was the first king of Qing dynasty in China), the characters of Manchu language were made by using the Mongolian letters during social development. When the Qing dynasty came to China, the Manchu language became the national language and Manchu characters became the written language. Therefore, there were many official files written with Manchu characters. Nowadays, there are over two million Manchu characters file in Chinese No.1 historical archives museum; documents contain the policy, economy, culture, military affairs, astronomy and atmosphere, etc. of Qing dynasty. It is highly valuable in history. Because the Manchu language has become obsolete, there are very few experts on it; most history research departments have made slow progress in researching these historical materials. Therefore, it is very important to develop offline handwritten primitive Manchu character recognition system.

Through the offline handwritten primitive Manchu character recognition system; researchers can scan all kinds of primitive Manchu documents in the computer and convert them to Manchu-roman code. Based on this, some office application can be done. Manchu-roman code is a Latinization form of primitive Manchu[1][2].

At present, printed character recognition system and online handwritten character recognition system of Chinese characters have come into a practical stage, the products of offline handwritten English and numerical recognition system have matured, but the primitive Manchu character recognition method research has never been reported either in China or abroad.

The application of offline handwritten primitive Manchu character recognition system is very important for researching the Qing dynasty's history. It also has important value for the other language characters recognition that belongs to Altai language family, especially the Mongolian characters and Xibo characters recognition research.

2. The features of Manchu

Manchu language is a branch of Altai phylum, a family of Tungus languages. In 1599, Nuerhachi ordered Erdeni and Gegai to create the Manchu language based on Mongolia language. This kind of language was written in a vertical direction with no space between the letters. Because it has no circle and dot, it was called "Primitive Manchu language without circle and dot" or "Old Manchu language". It was only used for 20 years. In 1632, Dahai improve it by adding circle and dot, standardizing letters and distinguishing the pronunciation. Then it was named

Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004)
0-7695-2187-8/04 $20.00 © 2004 IEEE

"Primitive Manchu language with circle and dot" or "New Manchu language", we called it "Primitive Manchu" for short.

Because the primitive Manchu has no standard characters, Manchu-roman code was developed in order to provide information communication. It transfers primitive Manchu to Latin code, similar to the phonetic system of the Chinese language.

Figure 1 is part of Manchu-roman code and primitive Manchu character mirror table. From Figure 1 we can find that primitive Manchu's letter writing is divided into single-writing, head-writing, mid-writing and tail-writing - four kinds of writing forms (some letters have only two kinds) according to their different locations in a word.

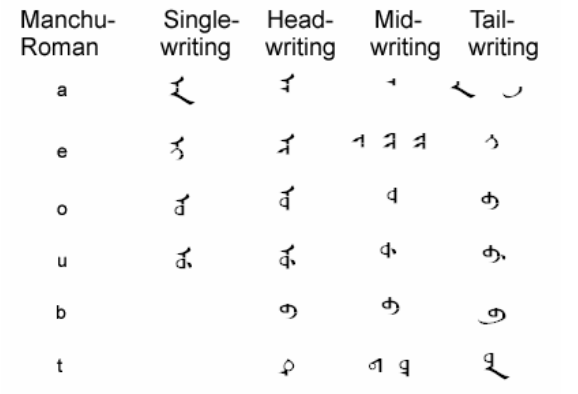

Figure 1. **The mirror table between Manchu-roman character and primitive Manchu character (part)**

Manchu characters are one of the most complex characters, with many big differences compared with Chinese and English, such as construction, spelling and composition of word. Therefore, the developed methods in Chinese and English are little help for the construction of Manchu language information system.

Manchu language is a kind of spelling language[3]; it has a lot of similarities and differences with English:

• Manchu is made up of 40 basic letters, including 6 vowels and 36 consonants. The vowel can appear at any place; the consonant can appear alone or collocate with the vowels and the consonants' location is limited. For one single letter, it has eight kinds of writing forms at most. Different letter collocation and different letter location in a word can lead to different writing forms, so the total number of letter writing forms is 114 (very few of the writing forms are repeated).

• The order of writing form is from top to bottom and from left to right.

• Each Manchu word is made up of one or more Manchu letters, but there is no space between the letters.

• The width of each Manchu word is same, but the height is quite different.

There is a primitive Manchu word in figure 2.

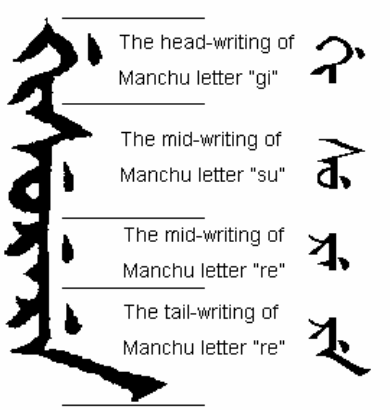

Figure 2. **Primitive Manchu word and letters**

3. System Sample

Primitive Manchu character recognition system principal diagram is in figure 3.

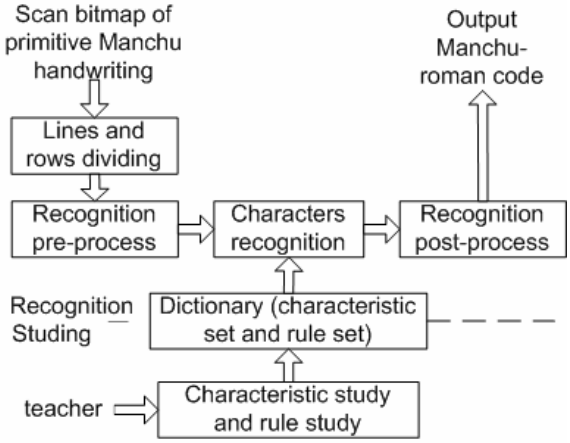

Figure 3. **Recognition system principal diagram**

According to figure 3, after the handwritten primitive Manchu character has been scanned into computer, it was digitalized as a binary image. A simple method is to be used to divide columns and rows after edge detection and lean correction in the typeset analysis. That is to scan columns one by one from column start point to its end and count the black pixel number at the same time. According to the result, words of primitive Manchu can be abstracted by dividing a threshold using a statistics method. Recognition pre-process[4][5] includes the following operations: thinning, pruning and noise reduction etc. When characters need to be recognized, words of primitive Manchu are divided into strokes. Then the stroke features are compared with sample stroke features, which have been stored in the computer, the most similar sample stroke is the result, the recognized stroke sequence was transformed into Manchu-roman code by using a compound recognition method. Recognition post-process includes processing the object with language

acknowledgment, looking for mistakes and auto-correcting them[6].

4. Word recognition

From figure 2, we can see, both words of primitive Manchu and English are composed of letters, but in a word of primitive Manchu, there is no space between the letters and letters have different written form in different position, which is opposite to English words. It is exceedingly difficult to distinguish letters in primitive Manchu words.

It is also impossible to recognize primitive Manchu words as a whole entity. First, because different primitive Manchu words have different height, which makes one word of Manchu very difficult to process; second, the quantity of expanded primitive Manchu words is over 200 thousand, which is an enormous quantity if it is categorized as a whole.

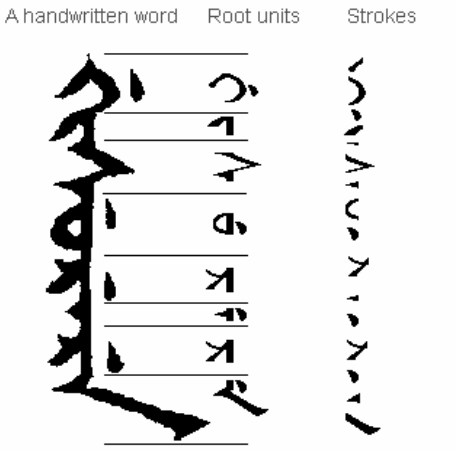

Figure.4 **Two-level sub-structure and strokes sample of a primitive Manchu word**

So we see that, a substructure of primitive Manchu characters needs to be constructed, which must be easy to recognize and can rebuild primitive Manchu character by rules. After research, all the written forms have been divided into 65 root units by comprehensive analysis of their common features. Then the root units were divided into 20 strokes. Under the strokes combined rule, the recognized stroke sequence makes up the units of root, which transform into Manchu-roman code[7][8]. The example of the two-level sub-structure is in figure 4. Experiments have proved that the strokes can be easily recognized after the strokes samples are studied. According to the combined rule, sequence of root units can be pieced together into Manchu-roman code. However, the test result showed many combination mistakes occurred even if there was little interference. So far, more rules that are exact are needed to standardize the combination of strokes. However, it is easy to make mistakes in the process of abstracting the rules of strokes.

In this case, the root units were canceled, the compound features of recognized strokes were contrasted with the stroke features in the library using fuzzy recognition method and Manchu-roman code was recognition result.

4.1 Recognition pre-process

Recognition pre-process plays an important role in the whole system, transforming patterns into the input form acceptable by the recognition system, eliminating noise. Recognition pre-process can accomplish the following functions:

• Noise reduction:

Morphologic noise filter is employed here. The equation (1) is a typical morphologic noise filter.

$$D = \{[(A\Theta B)\oplus B]\oplus B\}\Theta B = (A\circ B)\bullet B \qquad (1)$$

In this equation, A is the object to process; B is a structure element, which often uses 4-neighbourhood matrix or 8-neighbourhood matrix. D is the processed result.

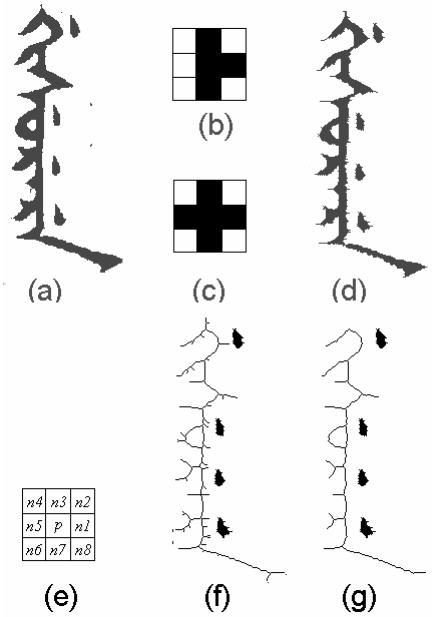

Figure 5. **Recognition pre-process. (a)primitive word; (b)corrupt structure element; (c)expanded structure element; (d) noise reduction result; (e)8-neighbourhood of pixel p; (f)thinning result; (g) pruning result**

The primitive Manchu written characters have more detailed features in the vertical left direction than in the vertical right direction. To adapt this writing feature and have a better reduction of noise, the equation (2) of the noise filter is used here. Structural element B is (b) in figure 5 and C is (c) in figure 5.

$$D = (((((A \Theta B) \Theta B) \oplus C) \oplus C) \Theta B) \Theta B \quad (2)$$

In figure 5, (d) is the result from (a) after the noise filter process.

•Thinning Main Body:

After the noise filter process, thinning object is needed to make process convenient. The algorithm of thinning is Hilditch thinning algorithm, which is described as following:

Suppose p is a pixel which needs to be processed, $f(p)$ is gray value of p, the object is a binary image, $n_i (i = 1,2,\ldots,8)$ is 8-neighbourhood pixel of p and the position of n_i listed at (e) in figure 5. Meanwhile, suppose set $I = \{1\}$ is the pixel subset that needs to be thinned, set $N = \{g | g - m \le 0\}$ is subset of background pixel, set $R = \{-m\}$ is the pixel changed from I after thinning m times.

Thinning conditions are:

(a) $f(p) \in I$;

(b) $U(p) \ge 1$, in which $U(p) = a_1 + a_3 + a_5 + a_7$;

(c) $V(p) \ge 2$, in which $V(p) = \sum_{i=1}^{8} (1 - a_i)$ and

$$a_i = \begin{cases} 1 & f(n_i) \in N \\ 0 & others \end{cases};$$

(d) $W(p) \ge 1$, in which $W(p) = \sum_{i=1}^{8} c_i$ and

$$c_i = \begin{cases} 1 & f(n_i) \in I \\ 0 & others \end{cases};$$

(e) $x(p) = 1$, in which $x(p) = \sum_{i=1}^{4} b_i$ and

$$b_i = \begin{cases} 1 & f(n_{2i-1}) \in N \wedge f(n_{2i}) \in I \cup R \vee f(n_{2i+1}) \in I \cup R \\ 0 & others \end{cases};$$

(f) $f(n_i) \notin R$ or $x_i(p) = 1, (i = 3,5)$, in which $x_i(p)$ refers to the value $x(p)$ which is I-neighborhood pixel of p.

In figure 5, the thinning result of (d) is (f).

•Pruning

From (f) of figure 5 we can see, some unnecessary branches have been generated in the process of thinning because of the tremble edge of the object, which should be removed by pruning. If the branch length that is calculated from the end to the branch point separation is less than a certain threshold, it should be deleted. The threshold value is given by equation (3).

$$d_c = \min(l_l, l_r)/4 \quad (3)$$

l_l means distance from the mid-axis of a word to its left boundary.

l_r means distance from the mid-axis of a word to its right boundary.

(g) in figure 5 is the result after pruning. It shows that pre-process result can reflect the basic features of the original character, and it is easy to get object features. During pre-process, disconnected strokes are not treated. There are only two kinds of disconnected strokes, so it is not necessary to treat them in thinning process.

4.2 Strokes separation

According to analysis, there are some rules for strokes to compound a primitive Manchu word. Every word has a mid-axis as its own main trunk; all the strokes extend from the main trunk. We can column-scan the original word image, and choose the column with the most significant pixel as mid-axis of the word. Different strokes can be separated into left strokes and right strokes according to whether they are on the left or right side of the main trunk, connected strokes and disconnected strokes according to whether they are connected with the main trunk or not. So all strokes can be separated into four categories, they are left disconnected strokes, left connected strokes, right disconnected strokes and right connected strokes, a total of 20 strokes ("circle" stroke is connect with main trunk when writing). Left disconnected strokes and right disconnected strokes have only one stroke, (left point and right point, easy to recognize), so the point of recognition is on left connected strokes and right connected strokes.

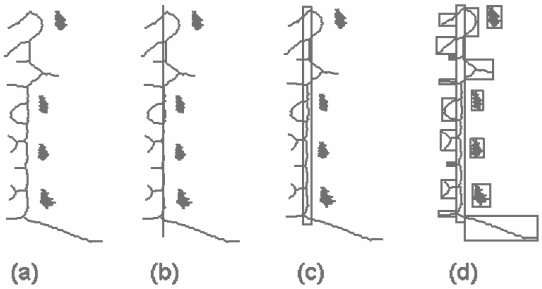

Figure 6. **Stroke separation. (a)pre-process result for primitive Manchu word; (b)confirm the mid-axis; (c)extend mid-axis; (d) stroke separation**

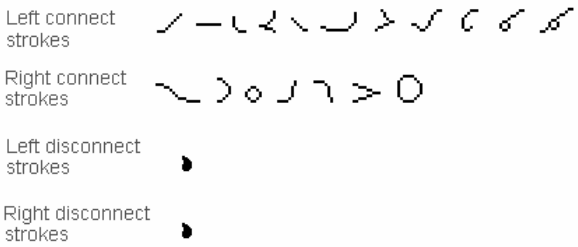

Figure 7. **Strokes categories**

Figure 6 is an example of stroke separation of Manchu words and Figure 7 is stroke categories.

4.3 Stroke recognition

After stroke separation, we need to get the feature code of the stroke. Here, the feature code is made up of 8-direction code[9] of the stroke. According to two points on each stroke, the direction code is given in equation (4).

$$k = (Y_{pt} - Y_{p(t-m)})/(X_{pt} - X_{p(t-m)}) \quad (4)$$

k means gradient value of two points, 8-direction code is decided by the gradients. m means number of the pixels between the two points to calculate direction code. If m is set too small, it is hard to eliminate noise of vibration. If m is set too large, detail information of the stroke would be lost. According to experimental result, we decided a three group's direction code of stroke:

•Left 8 feature code: From starting point of stroke (nearest point to mid-axis), trace the stroke by left branch preferential, until the first end point, dividing the pixel point sequence into 8 equal parts, then calculate the direction code between the starting point and the end point of the part one by one[10].

•Right 8 feature code: From starting point of stroke (nearest point to mid-axis), trace the stroke by right branch preferential, until the first end point, dividing the pixel point sequence into 8 equal parts, then calculate the direction code between the starting point and the end point of the part one by one.

•Left 16 total feature code: From starting point of stroke (nearest point to mid-axis), trace the stroke by left branch preferential, and all the end points are treated as branch points, until back to starting point, dividing the pixel point sequence into 16 equal parts, then calculate the direction code between the starting point and the end point of the part one by one.

The distance between the three feature codes and the relative feature codes in the feature library of the strokes is given by equation (5), and the results are represented by D_1、D_2、D_3. The feature library of the strokes is obtained by study[11][12].

$$D(X,G) = \sum_{i=1}^{m} |x_i - g_i| \quad (5)$$

In this equation, X means the feature code of input stroke; G means feature code of one standard stroke stored in the feature library of the strokes.

At last, we take the stroke relative to $\min(D_1 + D_2, D_3)$ as the result of stroke recognition.

4.4 Combination recognition

List the recognized strokes by their position in the word from top to bottom, that is the sequence of strokes of a word, through comparison with each primitive Manchu word standard stroke sequence in the library and then calculate the edit distance, the Manchu-roman code which has shortest distance is selected as final result, and a certain scope of candidate words is filtered as output. Edit distance describes the fewest steps to operate from string x to become y demands. Here, basic operation includes exchange operation, insert operation and delete operation. The edit distance calculation is given by equation (6) [13].

$$d = \sum_{i=0}^{m}\sum_{j=0}^{n} C[i,j] \quad (6)$$

In this equation,

$$C[i,j] = \min[C[i-1,j] + W_1, C[i,j-1] + W_2, C[i-1,j-1] + W_3 - \delta(x[i],u[j])].$$

C is an $m \times n$ integer cost matrix in this equation. $\delta(x[i],u[j])$ is expanded Kronecker δ function; it takes 1 if one variable provided is matching another variable; vice versa, it takes 0. $C[i-1,j]+1$ means insert cost; $C[i,j-1]+1$ means delete cost ; $C[i-1,j-1]+1-\delta(x[i],u[j])$ means exchange cost.

In the process of being recognized, a completely primitive Manchu word comes into a stroke sequence, which will be recomposed into five new strings $T_i(i=1,\ldots,5)$. Among T_i, T_1 is total strokes sequence; T_2 is left strokes sequence; T_3 is left feature strokes sequence; T_4 is right strokes sequence; T_5 is right feature strokes sequence. In library, each primitive Manchu word has corresponding standard strokes sequences $S_i(i=1,\ldots,5)$. By equation (6), edit distances between T_i and S_i is calculated. The results are $d_i(i=1,\ldots,5)$, which is the input of equation (8). In equation (8), insert weight value W_1, delete weight value W_2 and exchange weight value W_3 are decided by equation (7). Left strokes sequence is all strokes sequence that locates on the left side of mid-axis, including left disconnected strokes. So does the right strokes sequence. Left feature strokes and right feature strokes are those strokes clearly in the stroke recognition process. By experiment, zero left feature strokes and six feature strokes were defined.

$$W_1 = W_2 = W_3 = \begin{cases} 1 & i = 1 \\ 2 & others \end{cases} \quad (7)$$

Final combination distance d_r is calculated by equation (8).

$$d_r = \frac{(d_1 + d_2 + d_4 + (d_3 + d_5) \times 4)}{W_a{}^2} \quad (8)$$

Where, W_a is similarity of stroke sequence features, the features respectively are: total strokes sequence length, left disconnected strokes sequence length, right disconnected strokes sequence length, left feature strokes sequence length, right feature strokes sequence length, left strokes sequence length and right strokes sequence length. Recognition feature sequence is $l_{ti}(i=1,2,\ldots,7)$ and standard feature sequence is $l_{si}(i=1,2,\ldots,7)$ in library. Feature similarity is calculated by equation (9).

$$W_a(l_t,l_s)=\frac{\sum_{i=1}^{M} l_{ti}\times l_{si}}{\sqrt{(\sum_{i=1}^{M} l_{ti}{}^2)(\sum_{i=1}^{M} l_{si}{}^2)}} \qquad (9)$$

In this system, the primitive Manchu word, which has smallest combined distance d_r, is picked up as first candidate word and the candidate word list number is 10.

5. Conclusion

The offline handwritten primitive Manchu character recognition is a brand-new domain in handwritten character recognition. This paper gives a new method on how to recognized primitive Manchu words. This method combines the statistical pattern recognition method and the structural pattern recognition method. In the test, the source of study sample is the training of standard strokes, and there are 2597 word items' features stored in the library. The test samples come from a history file of Qing dynasty which name is "Gao Zong Shi Lu". This file had been written by hard coal staff and the strokes of word are standard. One hundred thirty-four words were selected random and twenty-three words were repeated. These test samples include all strokes. Rate of refuse recognition is 0%. Under the environment of Pentium II, the average recognition speed is 3.75 characters per minute. Once accurate recognition rate is 91.05%, in this test, one character was unrecognized because of the gravity noise and three were unrecognized because of the much irregular writing strokes (break in middle), except those disturb by noise and irregular strokes, the once accurate recognition rate is 93.85%. By analyzing, the irregular writing strokes are the main reason of being unrecognized. The test result shows that the proposed method is reasonable in recognizing Manchu handwriting characters. It should improve in recognition speed and anti-noise ability.

Acknowledgements

This project has gained the financial support from Natural Science Foundation of Liaoning Province and the support of Liaoning archives Library.

References

[1] Zhang G.Y., Li J.J., and Zhang L., "Implementation of Vector Database of Original Manchu Characters and Manchu/ Roman Characters Transform Inputting", *Journal of Northeastern University(Natural Science)*, Vol. 24, No. 11, Nov. 2003, pp. 1033-1036.

[2] Zhang G.Y., Li J.J., and Zhang L., "Realization of Language Interconversion Algorithm Between Roman-Manchu and Primitive Manchu", *Journal of Northeastern University (Natural Science)*, Vol. 24, No. 12, Dec. 2003, pp. 1157-1160.

[3] Qu L.S., *Manchu textbook*, Xinjiang People's Publishing House, Urumchi, 1991.

[4] S.W. Lee, "Offline recognition of totally unconstrained handwritten numerals using multilayer cluster neural network" *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 18, No. 6, Jun. 1996, pp. 648-652.

[5] Cai J., "Integration of structural and statistical information for unconstrained handwritten numeral recognition" *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 21, No. 2, Feb. 1999, pp. 263-267.

[6] E. Reiter, R. Dale, "Building applied natural language generation systems" *Natural Language Engineering*, Vol. 3, No. 1, Mar. 1997, pp. 57-87.

[7] Wu Z, R.T. Leahy, "An optimal graph theoretic approach to data clustering: theory and its application to image segmentation" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 11 Nov. 1993, pp. 1,101-1,113.

[8] S.B. Cho, "Neural-Network Classifiers for Recognizing Totally Unconstrained Handwritten Numerals", *IEEE Trans. PAMI.*, Vol. 8, No. 1, Jan. 1997, pp. 43-52.

[9] G. Srikantan, S.W. Lam, S.N. Srihari, "Gradient-based contour encoding for character recognition", *Pattern Recogn.*, Vol. 29, No. 7, Jul. 1996, pp. 1147-1160.

[10] T.M. Mitchell, *Machine Learning,* China Machine Press, Beijin, 2003.

[11] J.A. Bendediktsson, "Consensus theoretic classification methods", *IEEE Trans. Syst. Man Cybernet.*,Vol. 22, No. 4, Apr. 1992, pp. 688-704.

[12] T.K.Ho, J.J. Hull and S.N.Srihari, "Decision combination in multiple classifier systems", *IEEE Trans. Pattern Anal. Mach. Intell.*,Vol. 16, No. 1, pp. 66-75, Jan. 1994.

[13] Richard O.Duda, *Pattern Classification*, China Machine Press, Beijin, 2003.