



On-line Chinese character recognition using ART-based stroke classification

Hang Joon Kim ^{a,*}, Jong Wha Jung ^a, Sang Kyoong Kim ^b

^a Department of Computer Engineering, Kyung Pook National University, Taegu, 702-701, South Korea

^b Department of Computer Science, Inje University, Kimhae, 621-749, South Korea

Received 1 January 1996; revised 19 June 1996

Abstract

This paper proposes an on-line Chinese character recognition method using Adaptive Resonance Theory (ART) based stroke classification. Strokes, primitive components of Chinese characters, are usually warped into a cursive form and classifying them is very difficult. To deal with such cursive strokes, we consider them as a recognition unit and automatically classify them using an ART-2 neural network. The neural network has the advantage of assembling similar patterns together to form classes in a self-organized manner. This stroke classifier contributes to high stroke recognition rate and less recognition time. A database for character recognition also dynamically constructed with generalized character lists, and a new character can be included simply by adding a new sequence to the list. Character recognition is achieved by traversing the Chinese character database with a sequence of recognized strokes and positional relations between the strokes. To verify the performance of the system, we tested it on 1800 basic Chinese characters used daily in Korea, and obtained a good recognition rate of 93.13%. These results suggest that the proposed system is pertinent to be put into practical use.

Keywords: Character recognition; Adaptive Resonance Theory; Generalized character list; Fuzzy membership function

1. Introduction

With the advance of hardware technology, much research has been done on on-line character recognition to realize a more natural and accurate human-machine interface. We need to understand the characteristics of the target character set to develop a recognition method for it. Unlike the characters of Western alphabets such as English, Russian, French, German, some Oriental characters, such as Japanese, Korean and Chinese, have structural characteristics. That is, they are composed of smaller units in a

two-dimensional space. As a result, their character set is very large and they have many problems not found in Western alphabets.

Among these structural characteristics, Chinese has the following inherent characteristics:

- (1) Chinese is a large-alphabet language, which means that there are a great amount of categories;
- (2) there are many similar characters among different categories, so that a recognizer needs a more accurate recognition algorithm to discriminate correctly between similar characters;
- (3) on-line Chinese characters have wide variations in the number of strokes, stroke order and shape, and a recognizer should consider them.

* Corresponding author. E-mail: kimhj@bh.kyungpook.ac.kr.

We can classify on-line recognition methods according to the recognition unit, such as character (Chen and Chou, 1994), line segments of stroke (Hanaki and Yamazaki, 1980; Ikeda et al., 1981; Lin et al., 1993; Odake et al., 1982), stroke (J.W. Kim et al., 1995). Recognizing a character as a unit requires excessive amounts of memory space and processing time. To deal with these problems, a large-classification technique was used, usually resulting in an incomplete algorithm with many errors.

The straight line segment is a good recognition unit if a character is composed mainly of straight strokes. However, for recognizing cursive characters, this approach is not so efficient. Since there can be many variations of line segments for a given stroke, a cursive stroke tends to have many different representations. Hand-printed characters, having no stroke connections, consist of some basic straight strokes. Several researchers have reported a good recognition rate (Hanaki and Yamazaki, 1980; Ikeda et al., 1981); however, this method has the severe limitation that strokes should be disconnected in writing Chinese characters. Furthermore, on-line Chinese characters have many similar characters and classifying them depends heavily on correct recognition of cursive and connected strokes. If every cursive stroke is segmented into two or more straight segments, it becomes very difficult to classify similar characters. Thus, we have decided to consider a whole cursive stroke as a recognition unit (J.W. Kim et al., 1995; P.K. Kim and H.J. Kim, 1995; H.J. Kim and P.K. Kim, 1996; S.K. Kim et al., 1996), instead of using a small number of basic strokes and stroke segmentation techniques.

Although the number of Chinese characters is very large and the number of strokes per character varies dramatically, the basic shapes of such strokes are rather stable and limited to a certain number of categories because of writing behavior. So with an appropriate processing system, these strokes can be classified into a number of categories. Generally, conventional classification methods with supervised learning (Fan et al., 1995) need a set of pre-defined primitives such as basic strokes, line segments or curve primitives. In these approaches, because the paring of each input pattern with one of the pre-defined primitives is essential, manual effort is required to group the entire set of Chinese characters into

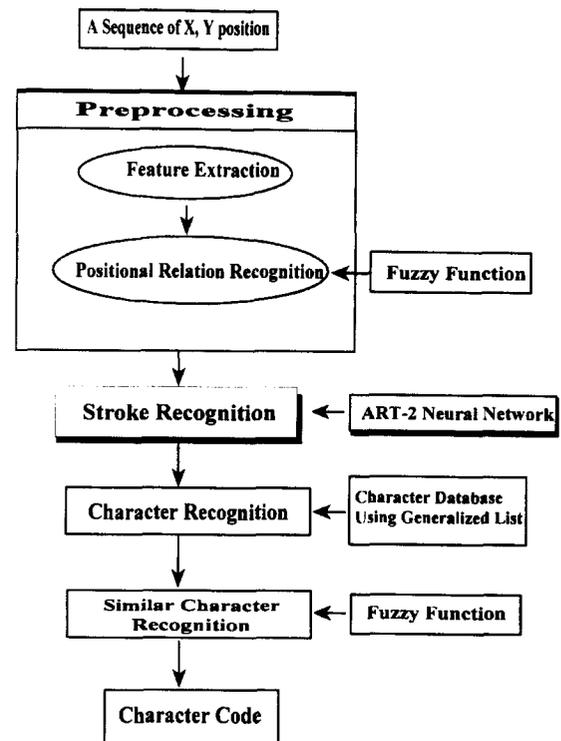


Fig. 1. The overall structure of the recognition system.

classes. This process is not only difficult but also prone to error. It would also be complex to build a classifier that directly recognizes each pattern. Therefore, we need a technique that

- (1) classifies input patterns into classes of similar patterns in self-organized manner;
- (2) directly activates a recognition code corresponding to an input pattern in real-time; and
- (3) accommodates the variability of input patterns with discrimination power.

In this paper, we propose a novel method for on-line recognition of Chinese characters using ART-based stroke classification. Fig. 1 shows the overall structure of the recognition system. To deal with cursive strokes, we use an ART-2 neural network. The available classes into which the numerous cursive strokes should be grouped are not easy to know in advance. To overcome this, we use an ART-2 neural network that automatically assembles similar patterns together to form classes in a self-organized manner. With a sequence of the classified strokes including their positional relations, character

recognition is achieved by traversing a character database learned with previous experience. The database consists of generalized lists which represent characters, and a new character can be simply learned by adding a sequence of strokes and positional relations for it. Although the self-organizing stroke classifier with the character database can deal with many recognition problems such as writing style, character size, stroke order, structural positioning, etc., there is a difficulty in discriminating similar characters. In order to tell similar characters apart, we use fuzzy membership functions.

This paper is organized as follows. We describe preprocessing in Section 2. The recognition of strokes and Chinese characters is illustrated in Sections 3 and 4, respectively. Experimental results in Section 5 are followed by conclusions in Section 6.

2. Preprocessing

2.1. Feature extraction

In this step, we make input vectors from input strokes for an ART-2 neural network to classify

cursive strokes. These vectors consist of normalized analog values that fall in the range between 0 and 1. Raw stroke data contain enormous variations in writing speed. Because these variations lead to poor performance, after normalization we resample the stroke for much better performance. The resampled stroke has 30 points. To obtain the vectors providing local information along the pen trajectory, we use pen coordinates, direction and curvature of the resampled point as features (Guyon et al., 1991). The features are pen coordinates (f_1, f_2), direction (f_3, f_4) and curvature (f_5, f_6) as shown in Fig. 2(c). These 180-dimensional feature vectors are the input vectors of the ART-2 stroke classifier. Fig. 2(d) shows an example of a feature vector from the stroke of Fig. 2(a). The time increases along the horizontal axis, from left to right. Each vector is represented as a column of boxes whose color indicates a value between 0 and 1.

2.2. Positional relation

To recognize a character correctly with identified strokes, we need to grasp their positional relations. For the purpose of computing the positional rela-

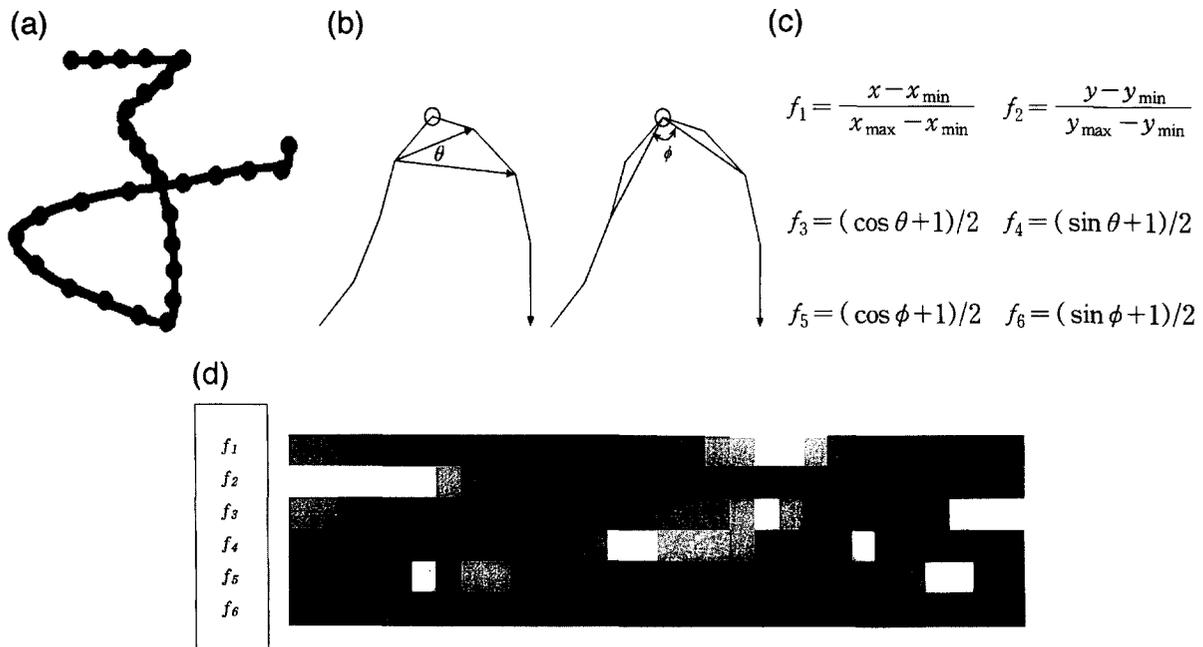


Fig. 2. Example of feature vector from an input stroke. (a) An input stroke. (b) Estimation of the slope and the curvature. (c) The features. (d) The feature vector of (a).

tions, we use five positional relations: *Up*, *Down*, *Left*, *Right* and *Cross*, which are represented by U , D , L , R and C , respectively. But these positional relations have some ambiguity. To overcome this ambiguity, we use fuzzy membership functions. The parameters of fuzzy membership functions for positional relations are the angle (θ) between the centers of two consecutive strokes and the distance (D) between the boundaries of two strokes. These two parameters are calculated by using the top-left point (Lx_i, Ty_i) and the bottom-right point (Rx_i, By_i) of the smallest rectangle which contains a stroke i ; and the minimum boundary center of the rectangle (Cx_i, Cy_i) (see Fig. 3(a)). As shown in Fig. 4(a), the five fuzzy functions are calculated by using angle, and Figs. 4(b)–(f) show the five fuzzy functions are calculated by using distance. In Fig. 4(f), A is the radius of the influence area which was surrounded by the minimum rectangle of two consecutive strokes. B is the radius of the influence area which was surrounded by the mean rectangle of two consecutive strokes. C is the radius of the influence area which was surrounded by the maximum rectangle of two consecutive strokes. D_s is the distance between the centers of two consecutive strokes. Fig. 5 shows an example calculation of the positional relation of input character 加 and 召. In this phase, we sum two fuzzy functions for each positional relation and choose the maximum value as the positional relation of two consecutive strokes.

3. Stroke recognition

Usually on-line Chinese characters have many cursive strokes which tend to have wide variations in their shape. Thus, handling of these cursive strokes is a very difficult problem. To overcome this problem, we consider cursive strokes as a recognition unit and automatically classify them using an ART-2 neural network. The significant feature of the ART-2 neural network is its ability to self-organize and self-stabilize its recognition codes in response to arbitrary sequences of analog inputs. In other words, it learns or adapts to new inputs while at the same time attempting to retain its previously learned information in some stable state. It also has the advantage that it is designed to learn quickly and stably in real

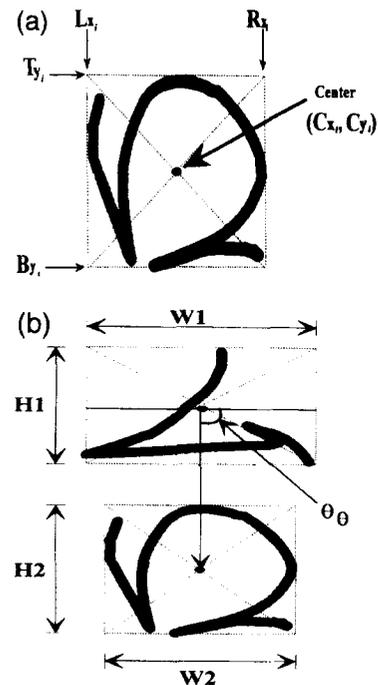


Fig. 3. Features used in fuzzy function. $W_i = Rx_i - Lx_i$ and $H_i = Ty_i - By_i$.

time in response to a changing world with an unlimited number of inputs until it runs out of memory. These properties reveal the good characteristics of the ART-2 neural network when it is used as a tool for learning recognition categories and recognizing patterns in real time.

Fig. 6 shows a typical ART-2 architecture. This ART-2 module includes the principal components of all ART modules, namely an attentional subsystem, which contains an input representation field F_1 and a category representation field F_2 , and an orienting subsystem, which interacts with the attentional subsystem to carry out an internally-controlled search process.

The two fields are linked by both a bottom-up adaptive filter and a top-down adaptive filter. A path from the F_1 node to the F_2 node contains a long term memory (LTM) trace, or adaptive weight.

Fig. 6 also illustrates the ART-2 feature, the three layers of the F_1 field and the preprocessing field F_0 . Each M -dimensional input vector I_0 is presented to the preprocessing layer F_0 and F_1 . Each layer of the F_0 and F_1 short-term memory (STM) fields carries

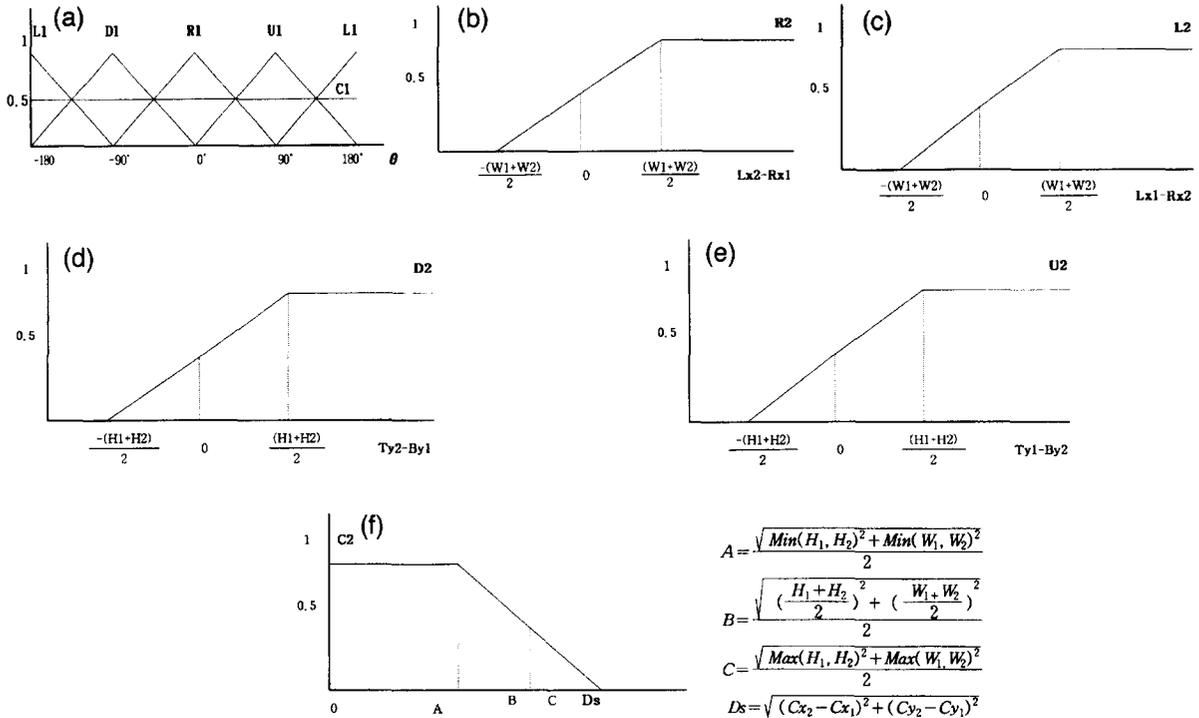


Fig. 4. Fuzzy membership functions for positional relations. (a) Fuzzy functions by angle (θ). (b) Fuzzy function by distance for the *Right* position. (c) Fuzzy function by distance for the *Left* position. (d) Fuzzy function by distance for the *Down* position. (e) Fuzzy function by distance for the *Up* position. (f) Fuzzy function by distance for the *Cross* position.

out two computations during the STM stabilization phase: intrafield and interfield inputs to that layer are summed and the resulting vector is then normalized by Euclidean normalization. In Fig. 6, the large filled circles represent normalization operations.

F_0 layer:

$$w_i^0 = I_i^0 + au_i^0, \tag{1}$$

$$x_i^0 = \frac{w_i^0}{e + \|w_i^0\|}, \tag{2}$$

$$v_i^0 = f(x_i^0) + bf(q_i^0), \tag{3}$$

$$u_i^0 = \frac{v_i^0}{e + \|v_i^0\|}. \tag{4}$$

The non-linear thresholding function f in Eq. (3) is

$$\begin{aligned} f(x) &= 0 \quad \text{if } 0 \leq x < \Theta, \\ f(x) &= x \quad \text{if } x \geq \Theta. \end{aligned} \tag{5}$$

The threshold Θ is critical to the contrast-enhancement and noise-suppression functions of the STM field. Subthreshold signals are set to zero, while suprathreshold signals are amplified by the subsequent normalization step, Eq. (3), at the top F_0 layer. The threshold Θ is assumed to satisfy the following constrains:

$$0 < \Theta \leq \frac{1}{\sqrt{M}}. \tag{6}$$

char	pos. relation	Angle (θ)	Distance (D)	recognized pos. relation ($\theta + D$)
加	R	R1 1.00	R2 0.65	1.65
	L	L1 0.00	L2 0.00	0.00
	U	U1 0.01	U2 0.02	0.03
	D	D1 0.01	D2 0.02	0.03
	C	C1 0.50	C2 0.45	0.95
召	R	R1 0.01	R2 0.02	0.02
	L	L1 0.01	L2 0.01	0.02
	U	U1 0.00	U2 0.00	0.00
	D	D1 1.00	D2 0.64	1.64
	C	C1 0.50	C2 0.38	0.88

Fig. 5. Example calculation of positional relation for characters 加 and 召.

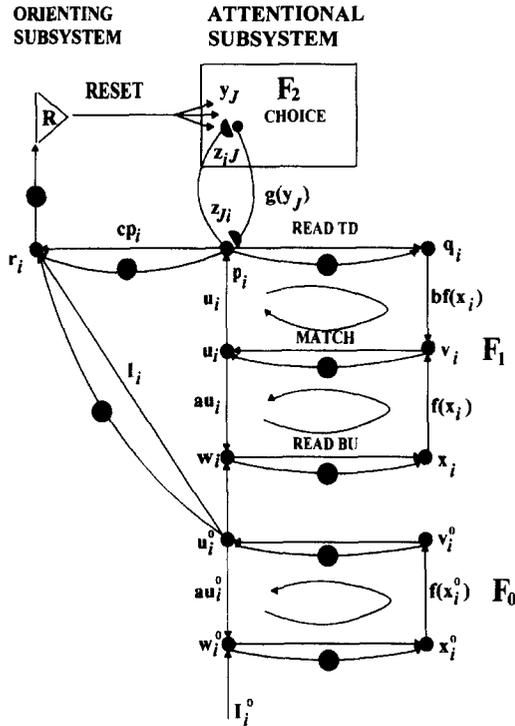


Fig. 6. ART-2 architecture (Carpenter et al., 1991).

The F_0 preprocessing stage is designed to allow ART-2 to satisfy a fundamental ART constraint; an input pattern must be able to instate itself in F_1 STM, without triggering reset, at least until an F_2 category representation becomes active and sends top-down signals to F_1 .

F_1 layer:

$$w_i = I_i + au_i, \quad (7)$$

$$x_i = \frac{w_i}{e + \|w_i\|}, \quad (8)$$

$$v_i = f(x_i) + bf(q_i), \quad (9)$$

$$u_i = \frac{v_i}{e + \|v_i\|}, \quad (10)$$

$$p_i = u_i + \sum_j g(y_j)Z_{ji}, \quad (11)$$

$$q_i = \frac{p_i}{e + \|p_i\|}. \quad (12)$$

As in F_0 , each F_1 layer sums inputs and normalizes the resulting vector. At the top F_1 layer p sums

both the internal F_1 signal u and the LTM trace, top-down from the j th F_2 node to the i th F_1 node Z_{ji} .

In the F_2 layer, the input from F_1 to the j th F_2 node is proportional to the sum $\sum_i p_i Z_{ij}$. When F_2 is designed to make a choice winner node, the j th node becomes active if

$$\sum_i I_i Z_{ij} = \max \left\{ \sum_i I_i Z_{ij} \right\}. \quad (13)$$

While the initial F_2 node selection is determined by Eq. (13), the orienting subsystem monitors the degree of match between the LTM trace pattern top-down of the chosen category and bottom-up input I . System reset occurs if

$$\|r\| < \rho, \quad (14)$$

where ρ is a dimensionless vigilance parameter between 0 and 1. Vector r obeys the following equation:

$$r = \frac{I + cp}{\|I\| + \|cp\|}, \quad (15)$$

where $c > 0$. Thus,

$$\|r\| = \frac{[\|I\|^2 + 2c\|I\|\|p\|\cos(I, p) + c^2\|p\|^2]^{1/2}}{\|I\| + c\|p\|}. \quad (16)$$

Once one F_2 node is reset, the reset node is committed and ART-2 activates the F_2 node J with the next highest input. If the maximally active F_2 node J matches the bottom-up input within the acceptance level, resonance develops. The bottom-up Z_{ij} and the top-down Z_{ji} LTM traces are updated as illustrated in the following equations (Carpenter and Grossberg, 1987, 1988; Carpenter et al., 1991; Gan and Lua, 1992):

$$Z_{ji} = \frac{I_i}{(1-d)}, \quad (17)$$

$$Z_{ij} = \frac{I_i}{(1-d)}. \quad (18)$$

The parameters a, b, c, d, e, ρ and Θ in the equations are ART-2 neural network parameters. In our implementation, $a = b = 10$, $c = 0.1$, $d = 0.8$ and $e = 0.0$. The vigilance parameter ρ determines the coarseness of the classification, that is, the measure of similarity between the input and the categorized templates. Hence, it is important to choose a level

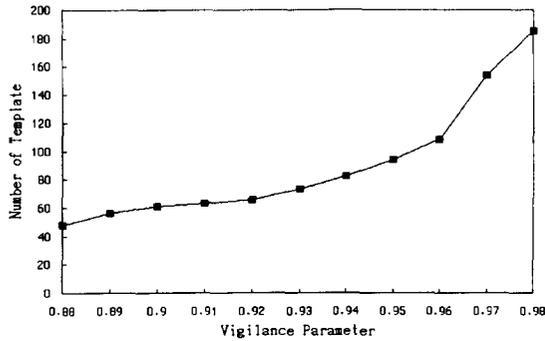


Fig. 7. The number of templates according to vigilance parameter.

appropriate for providing proper discernability in the classification. In the grouping of strokes with trained data, we initialize ρ to 0.88 and increase it by 0.01 each time while categorizing the input strokes with threshold parameter (θ) to 0.07. The number of templates varies for each level (see Fig. 7). Fig. 8 depicts templates when ρ is 0.92 and 0.96.

4. Character recognition

4.1. Database for character recognition

An efficient data structure is required to represent the large character set of the Chinese language. We construct dynamically a Chinese character database using a generalized list which can recognize a character by searching the database. The proposed recognition method using a database composed of generalized lists has three good features. First, character recognition is done in parallel with writing. The character list is traversed as a stroke is written and this enables fast recognition. Second, a new character can be learned to the list simply by adding the sequence of a new character. Third, storage for character representation is saved since many nodes share many parts of different characters. With the sequence, the database is automatically constructed. Fig. 10(a) shows the node structure of the list. A node has a stroke number, a positional relation, a possible character code and pointers. Characters are represented as illustrated in Fig. 9. Each character consists of recognized stroke number and positional relation between strokes. Fig. 10(b) represents part

of the character database which uses the generalized lists. This character list represents the “OR” relation between left-right nodes and the “AND” relation between upper-down nodes. This list is used in character recognition by searching the path from root

(a)

No	stroke shape						
1	一	2	丨	3	丿	4	㇇
5	㇇	6	㇇	7	㇇	8	㇇
9	㇇	10	㇇	11	㇇	12	㇇
13	㇇	14	㇇	15	㇇	16	㇇
17	㇇	18	㇇	19	㇇	20	㇇
21	㇇	22	㇇	23	㇇	24	㇇
25	㇇	26	㇇	27	㇇	28	㇇
29	㇇	30	㇇	31	㇇	32	㇇
33	㇇	34	㇇	35	㇇	36	㇇
37	㇇	38	㇇	39	㇇	40	㇇
41	㇇	42	㇇	43	㇇	44	㇇
45	㇇	46	㇇	47	㇇	48	㇇
49	㇇	50	㇇	51	㇇	52	㇇
53	㇇	54	㇇	55	㇇	56	㇇
57	㇇	58	㇇	59	㇇	60	㇇
61	㇇	62	㇇	63	㇇	64	㇇
65	㇇	66	㇇				

(b)

No	stroke shape						
1	一	2	丨	3	丿	4	㇇
5	㇇	6	㇇	7	㇇	8	㇇
9	㇇	10	㇇	11	㇇	12	㇇
13	㇇	14	㇇	15	㇇	16	㇇
17	㇇	18	㇇	19	㇇	20	㇇
21	㇇	22	㇇	23	㇇	24	㇇
25	㇇	26	㇇	27	㇇	28	㇇
29	㇇	30	㇇	31	㇇	32	㇇
33	㇇	34	㇇	35	㇇	36	㇇
37	㇇	38	㇇	39	㇇	40	㇇
41	㇇	42	㇇	43	㇇	44	㇇
45	㇇	46	㇇	47	㇇	48	㇇
49	㇇	50	㇇	51	㇇	52	㇇
53	㇇	54	㇇	55	㇇	56	㇇
57	㇇	58	㇇	59	㇇	60	㇇
61	㇇	62	㇇	63	㇇	64	㇇
65	㇇	66	㇇	67	㇇	68	㇇
69	㇇	70	㇇	71	㇇	72	㇇
73	㇇	74	㇇	75	㇇	76	㇇
77	㇇	78	㇇	79	㇇	80	㇇
81	㇇	82	㇇	83	㇇	84	㇇
85	㇇	86	㇇	87	㇇	88	㇇
89	㇇	90	㇇	91	㇇	92	㇇
93	㇇	94	㇇	95	㇇	96	㇇
97	㇇	98	㇇	99	㇇	100	㇇
101	㇇	102	㇇	103	㇇	104	㇇
105	㇇	106	㇇	107	㇇	108	㇇

Fig. 8. Categories with vigilance. (a) Categories when $\rho = 0.92$. (b) Categories when $\rho = 0.96$.

character	character representation	character	character representation
古	41 D 9 R 34	有	4 C 1 D 2 R 7 D 31 D 31
石	1 D 4 D 40	功	55 R 4 C 6
金	4 R 9 D 1 D 2 C 19 D 59	好	15 C 1 D 4 R 104
法	42 D 42 D 8 R 2 C 31 D 1 D 30	究	42 D 84 D 4 R 11 D 4 R 14
學	54 R 31 D 31 R 21 L 31 D 31 L 86 D 86 D 82 D 104		

Fig. 9. Examples of Chinese character representation.

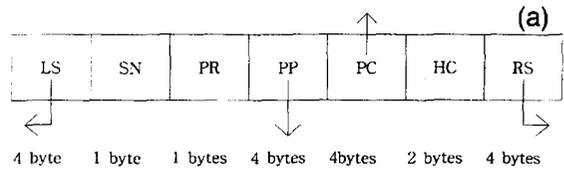
node to terminal node. In this case, we recognize the two character 古 and 十. It shows that this character list has the advantage of storage by sharing a few nodes between different characters.

4.2. Similar character recognition

Although the ART-2 strokes recognizer with the character database works well with cursive Chinese characters, it cannot discriminate perfectly between similar characters. We investigate these similar characters and use fuzzy functions to tell them apart. Table 1 shows a group of similar characters that are chosen as similar character recognition targets in our system. To discriminate between these similar char-

Table 1
The group of similar characters

- *2 [人 入 八], [刀 力], [十 丁]
- *3 [勺 久 夕], [丈 大], [执 凡], [巳 己 巴], [土 士 工]
- *4 [夬 尺], [夕 友], [牛 午] [日 口], [父 爻], [太 犬], [天 夫]
- *5 [血 皿], [甲 田 申 由 用], [夬 只 史], [生 乍], [且 目], [加 召], [未 末], [石 右]
- *6 [名 各], [有 百], [未 朱]
- *7 [沈 沉], [佃 伸], [邑 吧]
- *8 [朋 明]
- *9 [胃 胃]
- *10 [鈔 鈞], [細 紳]
- *11 [釉 番]
- *12 [鉀 鉅]



LS : Left Sibling
 SN : Stroke Number
 PR : Positional Relation
 PC : Point to a Child
 HC : Chinese Character Code
 RS : Right Sibling
 PP : Point to a Parent

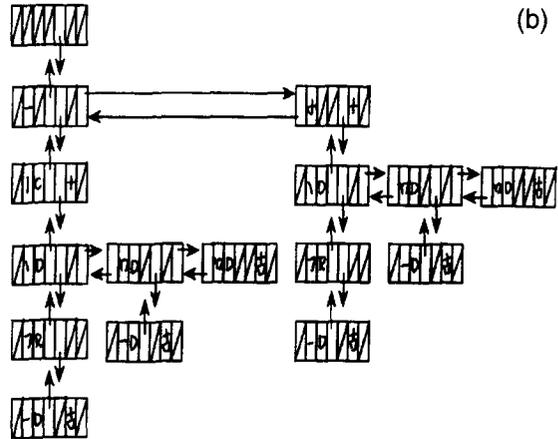


Fig. 10. The structure of the generalized character list. (a) The node structure of the generalized character list (Horowitz and Sahni, 1987). (b) Part of the character list (古 and 十).

acters, we construct fuzzy functions manually in each case. For example, Fig. 11(a) shows a fuzzy function by using the width rate of a stroke and Fig. 11(b) shows a fuzzy function by using the height per width rate of a character.

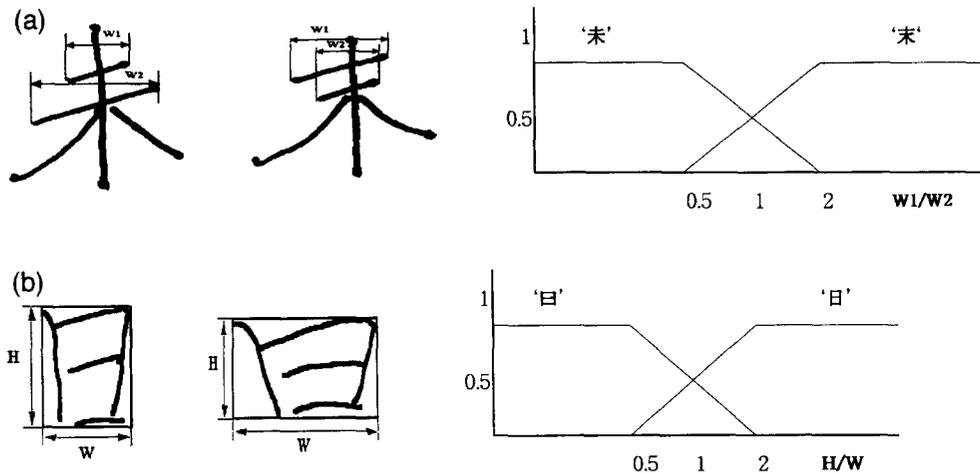


Fig. 11. Example of fuzzy function for similar characters. (a) Example of fuzzy function by using the width rate of a stroke. (b) Example of fuzzy function by using the height per width rate of a character.

5. Experimental results

To evaluate the practical performance of the proposed method, experiments were done on 10 sets of 1800 basic Chinese characters used daily in Korea. The data were written by 40 different writers without any constraints. WACOM HD-648A was used as an input device and the method was implemented in C++ language on an IBM PC-Pentium. In the stroke classification phase, about 76,000 strokes from four selected character sets were put into the ART-2 neural network, and classified into categorized groups according to a vigilance parameter. Fig. 12 shows the rate of correct stroke classification according to the vigilance parameter. As the level of vigilance increases, the success rate of the classification improves. However, the performance deteriorates gradually around 0.96. Because we determine the classification rate by using two criterion factors. The first factor is an intra-class density and the other factor is an inter-class isolation. In Fig. 12, the intra-class density increases according to the vigilance parameter, but beyond a vigilance parameter of 0.96, the inter-class isolation decreases radically. In other words, if the vigilance parameter is beyond 0.96, the same strokes are divided into different classes according to a very small difference. Thus, the overall correct classification rate is deteriorated when the value of the vigilance parameter is beyond 0.96.

We chose 0.96 as the vigilance parameter of the ART-2 neural network to classify strokes.

Table 2 shows the results of the experiment. The correct recognition rate of test characters except similar characters was 93.25% (α) and that of similar characters in the test characters was 88.76% (β). When the rate of similar characters from the test characters is γ , the overall recognition rate can be calculated as $(1 - \gamma) \times \alpha + \gamma \times \beta$. Thus, the overall recognition rate for the tested characters is 93.13%.

Recognition errors were mainly caused by abnormal writing, the use of unlearned strokes, unlearned characters, and ambiguous positional relations. Only a small percentage of strokes was mis-classified and this proves the usefulness of ART-based stroke classification. Most rejections in the test data were from

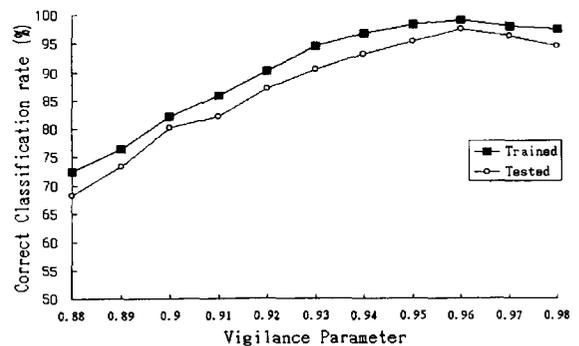


Fig. 12. Correct classification rate.

Table 2
Recognition rate

Char.	No. of char.	Correct recog.	Mis-recog.	Rejected	Similar char.	Similar char. recognition rate (β)	Recog. rate except similar char. (α)	Overall recog. rate
Train	7,200	6848	176	0	176	93.18	97.49	97.38
Test	10,800	9727	682	416	276	88.76	93.25	93.13

Table 3
Relative error ratio of test character

Error type	No. of error characters	Rate (%)
Abnormal writing	133	12.11
Unlearned stroke	397	36.15
Unlearned character	416	37.88
Ambiguous positional relation	152	13.84
Total	1098	100.00

unlearned characters. This can be overcome by more training in the unlearned characters. Table 3 shows the relative error ratio according to error types. Table 4 shows the comparison of on-line character recognition systems. This table shows that the recognition rate of the proposed system is not too bad in comparison with other systems. In a view of constraints in writing characters, other systems ask writers to write characters within a framework of constraints as to

the number and order of components and the size of the character. But our system has no constraint in writing characters. In this aspect, the proposed system has some advantage over other methods. Fig. 13 shows four examples that were mis-recognized. The character in (a) was mis-recognized as Ⅲ, because the first stroke was written abnormally. The character in (b) was written with only one cursive stroke in the dotted box, which was among the unlearned strokes and was unusual. In (c), the character 學 was written in an unusual manner, and was an unlearned character. Lastly, the character in (d) was mis-recognized because this character required a more accurate positional description. Fig. 14 shows recognition rates according to character sets. The first four character sets were chosen as training data and the others were used as test data. The graph shows even recognition rates for all test characters. This proves the effectiveness of the proposed recognition system. Sample test characters are shown in Fig. 15. Fig. 16 shows an implementation of the proposed method.

Table 4
Comparison of on-line character recognition system (Nouboud and Plamondon, 1990)

Author	Types of char.	Constraints	No. of writers	No. of characters	Recognition result (%)	Calculation time
Arakawa	Kanji, Hiragana Katakana and Alphanumeric	Comp. order Comp. number	35	35 specimens per char.	99.2 to 99.7	
Ikeda et al.	Kanji, Hiragana Katakana and Alphanumeric	Comp. number Comp. order		17,463	92 to 98	Real time
Odaka et al.	Kanji, Hiragana Katakana and Alphanumeric	Boxes Comp. number	40	9,135	99.5	Real time
H.J. Kim et al.	Chinese character	No constraints	40	18,000	93.13 to 97.38	Real time

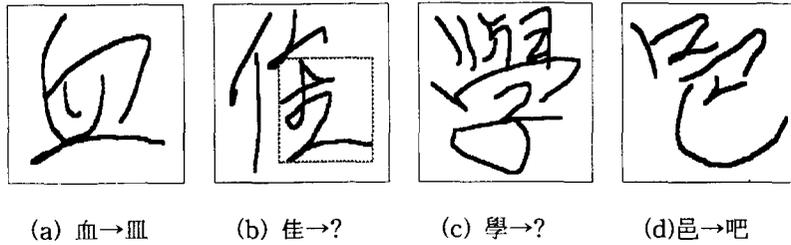


Fig. 13. Examples of recognition error.

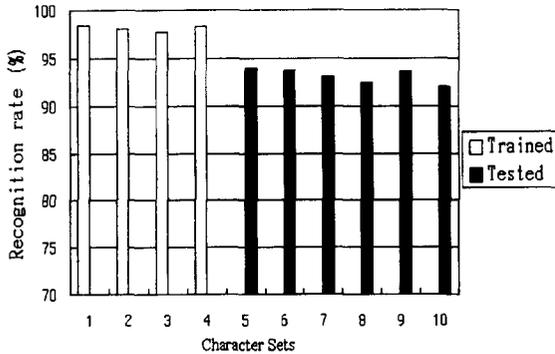


Fig. 14. Character recognition rate according to character sets.

6. Conclusions

Conventional research on the recognition of on-line Chinese characters has some difficulties in describing cursive strokes. To deal with these prob-

lems, we presented a new recognition method using ART-based stroke classification. We use whole cursive strokes as a recognition unit, and classify them by using an ART-2 neural network. For a correct and simple character representation, five positional relations between two consecutive strokes are defined. With an input sequence of recognized strokes and the positional relations, we recognize an input character by traversing a Chinese character database constructed with generalized character lists. Furthermore, to discriminate between characters from a group of similar Chinese characters, we use fuzzy membership functions.

Experimental results showed that the proposed method has the potential for recognizing cursive Chinese characters. Below are the advantages of the proposed method.

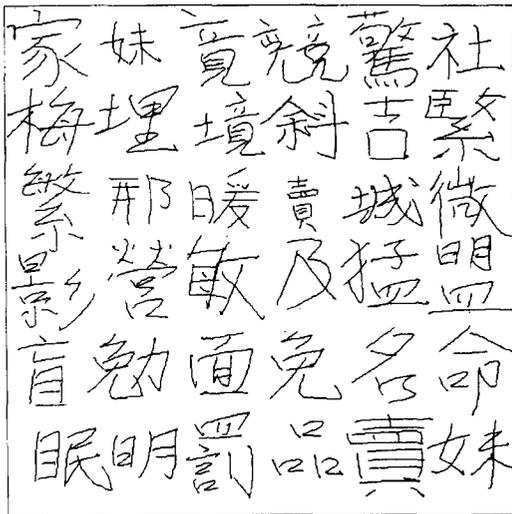


Fig. 15. Sample test characters.

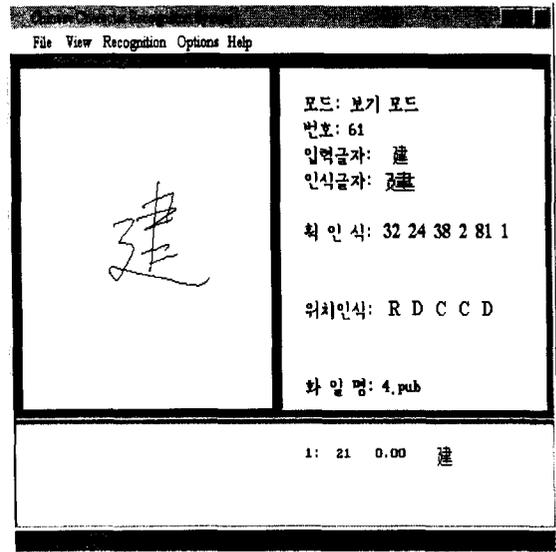


Fig. 16. System layout.

- (1) The recognition unit of whole cursive strokes does not suffer from stroke segmentation and stroke extraction.
- (2) Because the ART-2 stroke classifier can automatically classify input patterns into classes of similar patterns in a self-organized manner, manual definition of the actual strokes for real-world characters (which is a critical process for on-line character recognition) is unnecessary.
- (3) The ART-2 stroke classifier provides a very high recognition speed and a large degree of flexibility in dealing with distortion of strokes.
- (4) The Chinese character database with the generalized character lists has good extensibility and provides the ability to deal with variations of stroke order.
- (5) Fuzzy membership functions for similar Chinese characters have powerful discriminating ability.

Though the performance of the proposed method is very effective, it could be improved by more training data, which would result in a more practical recognition system. Future work will be concerned with the extension of the method to the whole Chinese character set. More specifically, in order to implement a large-scale system, a study of the character database and a retrieval strategy are necessary.

References

- Carpenter, G.A. and S. Grossberg (1987). A massively parallel architecture for a self organizing neural pattern recognition machine. *Comput. Vision Graphics Image Process.* 37, 54–115.
- Carpenter, G.A. and S. Grossberg (1988). The ART of adaptive pattern recognition by a self-organizing neural network. *IEEE Comput.* 21 (3), 77–88.
- Carpenter, G.A., S. Grossberg and D.B. Rosen (1991). ART2-A: an adaptive resonance algorithm for rapid category learning and recognition. *Neural Networks* 4, 493–504.
- Chen, W.-T. and T.-R. Chou (1994). A hierarchical deformation model for on-line cursive script recognition. *Pattern Recognition* 27 (2), 203–219.
- Fan, K.C., C.K. Lin and K.S. Chou (1995). Confusion set recognition of on-line Chinese characters by artificial intelligence technique. *Pattern Recognition* 28 (3), 303–313.
- Gan, K.W. and K.T. Lua (1992). Chinese character classification using an adaptive resonance network. *Pattern Recognition* 25 (8), 877–882.
- Guyon, I., P. Albrecht, Y. Le Cun, J. Denker and W. Hubbard (1991). Design of a neural network character recognizer for a touch terminal. *Pattern Recognition* 24 (2), 105–119.
- Hanaki, S.-I. and T. Yamazaki (1980). On-line recognition of handprinted Kanji characters. *Pattern Recognition* 12, 421–429.
- Horowitz, E. and S. Sahni (1987). *Fundamentals of Data Structures in Pascal*, 2nd Edition, 145–160.
- Ikeda, K., T. Yamamura, Y. Mitamura, S. Fujiwara, Y. Tominaga and T. Kiyonom (1981). On-line recognition of hand-written characters utilizing positional and stroke vector sequences. *Pattern Recognition* 13 (3), 191–206.
- Kim, H.J. and P.K. Kim (1996). On-line recognition of cursive Korean characters using a set of extended primitive strokes and fuzzy functions. *Pattern Recognition Lett.* 17, 19–28.
- Kim, J.W., K.C. Jung, S.K. Kim and H.J. Kim (1995). Shape classification of on-line Chinese character strokes using ART-1 neural network. In: *Proc. World Congress on Neural Network (WCNN'95)*, Vol. 2, II-191–II-194.
- Kim, P.K. and H.J. Kim (1995). On-line recognition of run-on Korean characters. In: *Proc. Internat. Conf. on Document Analysis and Recognition (ICDAR'95)*, Vol. I, 54–57.
- Kim, S.K., J.W. Kim and H.J. Kim (1996). On-line recognition of cursive Korean characters using neural networks. *NeuroComputing Lett.* 10, 291–305.
- Lin, C.-K., K.-C. Fan and F.T.-P. Lee (1993). On-line recognition by deviation-expansion model and dynamic programming matching. *Pattern Recognition* 26 (2), 259–268.
- Nouboud, F. and R. Plamondon (1990). On-line recognition of handprinted characters: survey and beta test. *Pattern Recognition* 23 (9), 1031–1044.
- Odaka, K., H. Arakawa and I. Masuda (1982). On-line recognition of handwritten characters by approximating each stroke with several points. *IEEE Trans. Syst. Man Cybernet.* 12 (6), 898–903.