Pergamon

Pattern Recognition, Vol. 30, No. 9, pp. 1489–1500, 1997
© 1997 Pattern Recognition Society. Published by Elsevier Science Ltd
Printed in Great Britain. All rights reserved
0031-3203/97 $17.00+.00

PII: S0031-3203(96)00161-6

ON-LINE RECOGNITION OF HANDWRITTEN CHINESE CHARACTERS BASED ON HIDDEN MARKOV MODELS

HANG JOON KIM,[†] KYUNG HYUN KIM,[†,*] SANG KYOON KIM[‡] and JONG KOOK LEE[§]

[†]Department of Computer Engineering, Kyung-Pook National University, Taegu, S. Korea
[‡]Department of Computer Science, Inje University, Kimhae, S. Korea
[§]Department of Computer Engineering, Andong National University, Andong, S. Korea

(*Received* 21 *March* 1996; *in revised form* 25 *September* 1996; *received for publication* 16 *October* 1996)

Abstract—Difficulties in Chinese character recognition due to numerous strokes usually warped into a cursive form and a much larger set of characters. In this paper, we propose a hidden Markov model (HMM) based recognition model that deals efficiently with these recognition problems. The model is an interconnection network of radical and ligature HMMs. It works well with variations of the cursive strokes by the characteristics of the HMMs. It represents the large character set with a relatively small memory and also has good extensibility. To solve the problem of recognition speed caused by a number of search paths, we combine a modified level building search with the isolated radical and ligature HMMs in an attempt to achieve a robust, accurate recognizer whose performance is optimized. The algorithm is an efficient network search procedure, the time complexity of which depends on the number of levels in the network. A test with 18,000 handwritten characters shows a recognition rate of 90.3% and a speed of 1.83 s per character. © 1997 Pattern Recognition Society. Published by Elsevier Science Ltd.

On-line Chinese character   Hidden Markov model (HMM)   Level building
Character recognition network

1. INTRODUCTION

Along with the development of hardware technology, many researches on on-line character recognition have been done to realize a more natural and accurate human–machine interface. However, the recognition of handwritten characters is still difficult due to a variety of shapes and the complexities of characters. Unlike the characters of Western alphabets such as English, Russian, French, German, etc., some Oriental characters, such as Korean, Japanese 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 is a great amount of categories; (2) on-line characters have many cursive strokes from wide handwriting variability among writers. The recognition of such cursive strokes and the representation of the large character set have an important effect on a practical recognition model of on-line Chinese characters.

From the view point of the recognition technique of on-line Chinese characters, there have been four major approaches:(1–3) (1) structural analysis approach; (2) statistical analysis approach; (3) neural network approach; (4) hybrid method or combination scheme. The neural network has been proven to be a powerful tool for recognizing static patterns such as off-line characters. To overcome the weakness of neural networks in handling temporal information of on-line characters, the error recurrent neural network can be introduced.(4) On the other hand, it would be too expensive in view of computation and complex to build a neural network classifier which directly recognizes each character of the large character set. The hybrid method of structural analysis and the artificial intelligence technique can complement each other in correctness.(3) This approach also worked well with similar Chinese characters. As more computing power is becoming available, this kind of approach is very promising. The structural analysis approach has been most popular and shown good results. Many useful matching techniques for the approach have been proposed, such as elastic matching, relaxation matching and dynamic programming matching.(5–7) Since these methods need a set of pre-defined primitives such as basic strokes, line segments or curve primitives, finding representative patterns that include the entire stroke set of Chinese characters is essential. This process is not only difficult but also prone to error. Moreover, they have the limitation that it is difficult to describe cursive strokes efficiently.

To deal with cursive strokes, we need a model which provides an explicit representation for time-varying input patterns and probabilistic interpretations accommodating variations in input patterns. HMMs have intrinsic properties that make them very attractive for handwriting recognition. They are: (1) HMMs use

[*] Author to whom correspondence should be addressed.

statistical algorithms that can automatically extract knowledge from training patterns, in contrast to knowledge-based approaches; (2) HMMs have modeling power that the patterns are implicitly modeled by different paths in the stochastic network, as compared to an explicit manner in most of the conventional approaches; (3) HMMs the have capability of naturally modeling temporal information.(9–16,20–24)

In this paper, we propose a recognition model of online cursive Chinese characters based on HMMs. The model is constructed as a character recognition network composed of radical and ligature HMMs supposing that a Chinese character is a set of consecutive and meaningful radicals. To improve recognition speed, we use a modified level building search procedure. The network represents information for the large character set: nodes describe the structural information of radicals which can be shared and reused by another character, and a path from the initial state to the final state represents a complete character. Therefore, the large character set can be represented by a small number of HMMs, and cursive characters are recognized in a way that accommodates variations in personal writing styles without stroke segmentation. In this model, character recognition is regarded as a network search which finds the optimal path with maximum probability. However, a network for the Chinese characters has so many paths to cover all the possible characters that an entire search cannot meet our expectation because of much computation requirement.

To resolve this problem, we use a level building technique. The level building algorithm(17–19) is an optimal network search procedure and we modify it for optimally time aligning a sequence of cursive Chinese strokes with a sequence of isolated HMMs as radical reference patterns. Instead of evaluating all existing paths, the algorithm once evaluates each HMM on a level using the Viterbi algorithm, then decides an appropriate radical on that level by backtracking. Therefore, the performance of the algorithm is governed only by the number of levels in the network, and is appropriate for the recognition network of Chinese characters.

The organization of this paper is as follows. In Section 2, HMM is reviewed briefly. In Section 3 and Section 4, a recognition network for Chinese characters and a network search procedure are presented, respectively. In these sections, we describe how to construct the recognizer and apply the level building search procedure. In Section 5, experimental results are illustrated. Finally, in Section 6, we present our conclusions and future work.

2. HIDDEN MARKOV MODEL

2.1. *The elements of HMM*

Since HMM was introduced at the end of 1960, it has been applied to connected, speaker-independent, automatic speech recognition with the advantage of modeling various patterns. Recently it has also been widely adapted to character recognition. HMM is represented by two interrelated mechanisms, an underlying Markov chain having a finite number of states, and a set of random functions, one of which is associated with each state. The elements of HMM are described as follows, using the same notation as used by Levinson *et al.*(9)

1. N: the number of states
2. M: the number of output symbols
3. T: the number of observations
4. $Q=\{q_t\}$: the set of states, $q_t \in \{1,2,\ldots,N\}$, $t = 1,2,\ldots,T$
5. $V=\{V_k\}$: the set of output symbols, $k = 1,2,\ldots,M$
6. $A=\{a_{ij}\}$: the transition matrix of the underlying Markov chain Here, a_{ij} is the probability of transiting to state j given current state q_i, that is
$$a_{ij} = P[q_{t+1} = S_j| = q_t = S_i], \quad 1 \le i,j \le N$$
7. $B=\{b_j(v_k)\}$: the model output symbol probability matrix, where $b_j(v_k)$ is the probability of output symbol v_k, given current state q_j
8. $\pi=\{\pi_i\}$: the initial state probability vector, $i = 1,2,\ldots,N$; $\pi_i = P[q_1 = i]$: the initial state probability given that the first state number is i

HMM can model the set of observations using these probabilistic parameters as a probabilistic function of an underlying Markov chain whose state transitions are not directly observable.

2.2. *A type of HMM*

The structure of HMM has various forms; an unconstrained model (ergodic model) in which a transition from any state to any other state can be made, and a left-to-right model in which as time increases the state index increases, that is, the states proceed from left to right. The later model has the following properties, so it is appropriate for modeling data having sequential variations as time goes on:(9)

1. No transitions are allowed to states whose indices are lower than the current state:
$$a_{ij} = 0, \quad j < i.$$
2. The state sequence must begin in state 1 and end in state N:
$$\pi_i = \begin{cases} 0 & i \neq 1, \\ 1 & i = 1. \end{cases}$$
3. Large changes in state index do not occur:
$$a_{ij} = 0, \quad j > i + \Delta.$$

There are two problems associated with this special HMM. One is that a single, long observation sequence is useless for training in such a model because once the state q_N is reached, the rest of the sequence provides no further information about earlier states. The other is that a scaling estimation procedure using a multiple observation sequence requires some care since the scale factors for each individual set of forward and backward probability are different. We explain modified training formulas solving these problems in the following section.

2.3. *Multiple observation sequence*

An important factor affecting the determination of the optimum parameters in the left-to-right model is the observation sequence used for training.(9) For constructing a writer-independent model, the observation sequence O consists of several independent sequences $O^{(k)}, k = 1, 2, \ldots, K$, where $O^{(k)}$ is the training sequence for writer k and K is the total number of writers used for training. Evaluating the probabilistic parameters using multiple training data is maximizing $P(O|\lambda)$ in the following equation under the assumption that each training sample is mutually independent:

$$P(O|\lambda) = \prod_{k=1}^{K} P(O^{(k)}|\lambda) = \prod_{k=1}^{K} P_k. \tag{1}$$

Because the re-estimation formulas for matrices A and B are based on frequencies of occurrence of various events, the new re-estimation formulas for multiple observation sequences are expanded by adding together the individual frequencies of occurrence for each sequence. Thus, the modified re-estimation formulas are

$$\bar{a}_{ij} = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_{k-1}} \alpha_t^k(i) a_{ij} b_j(O_{t+1}^{(k)}) \beta_{t+1}(j)}{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i) \beta_t^k(i)} \tag{2}$$

and

$$\bar{b}_j(l) = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1, s.t. O_t = v_l}^{T_k-1} \alpha_t^k(i) \beta_t^k(i)}{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k-1} \alpha_t^k(i) \beta_t^k(i)}, \tag{3}$$

where the function $\alpha_t^k(i)$ is $\Pr[O_1, O_2, \ldots, O_t$ and q_i at $t \mid \lambda]$, the function $\beta_t^k(j)$ is $\Pr[O_{t+1}, O_{t+2}, \ldots, O_T \mid q_i$ at t and $\lambda]$, and v_l denotes observing symbols. These functions are calculated by the forward–backward procedure,(8,10) and π_i is not re-estimated since $\pi_1 = 1$, $\pi_i = 0$, $i \neq 1$. This is a simple explanation of HMM, and we will adapt it to on-line recognition of cursive Chinese characters in the next section.

3. CHINESE CHARACTER RECOGNITION NETWORK

3.1. *Radical model and ligature model*

On-line handwriting is represented by a set of stroke sequences that are made up from points sampled by tracing between pen-down and pen-up. Pen-up ordinarily identifies position information between strokes. But when we write cursively, several continuous strokes are connected as a single stroke, and this information is not valid. These virtual lines are referred to as "ligature". Since it makes a lot of uncertainty, randomness and incompleteness, we choose HMM among various stochastic approaches which are known to deal with these problems efficiently. Then we have to choose a basic form of the model, such as the number of states and the kind of output symbols, in addition to probabilistic parameters generated by the training procedure for HMM. These choices about the given problems must be made depending on the experience of a designer. In this paper, we design basic models employing a left-to-right HMM, as shown in Fig. 1.

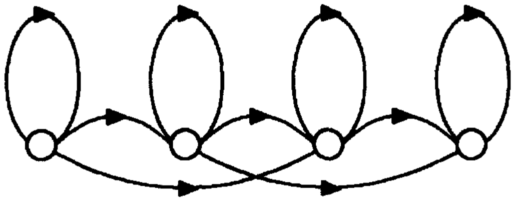


Fig. 1. A left-to-right HMM.

Generally, a complete character is able to be constructed by only one radical itself or by connecting the radicals in order of radical-ligature-radical, radical-ligature-radical-ligature-radical, and so on. Figure 2 shows that the character "穷" can be made by combining three radicals and two ligatures. Like this, we split each Chinese character into several radicals and ligatures. We represent the ligatures between strokes within radicals using radical HMMs and the ligatures between radicals using ligature HMMs separately. Then we allot the number of states in proportion to the length of the radical or ligature normalized. We use an eight-direction code for observable output symbols in each state and the Baum–Welch algorithm to train these data.

If the number of characters to be recognized is not numerous, we can construct one model corresponding to each complete character. But because the set of Chinese characters is very large, we need great memory capacity for maintaining each model and much computation for searching a suitable model, so that this approach is inadequate for real time applications.

The problem of requiring much memory is solved by the structural tying(15) in which the same shaped radicals in different characters are shared structurally. We connect a set of radical HMMs and ligature HMMs to make an efficient Chinese recognition model. The rule of construction is described by a network which defines all radical sequences which are considered valid characters. Figure 3 shows parts of all the network. Edges on the network depict the structure information of radicals which can be shared and reused by other characters. The same sort of Chinese characters on the different

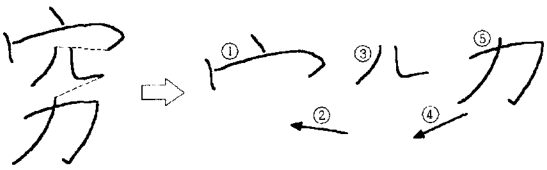


Fig. 2. Radicals and ligatures.

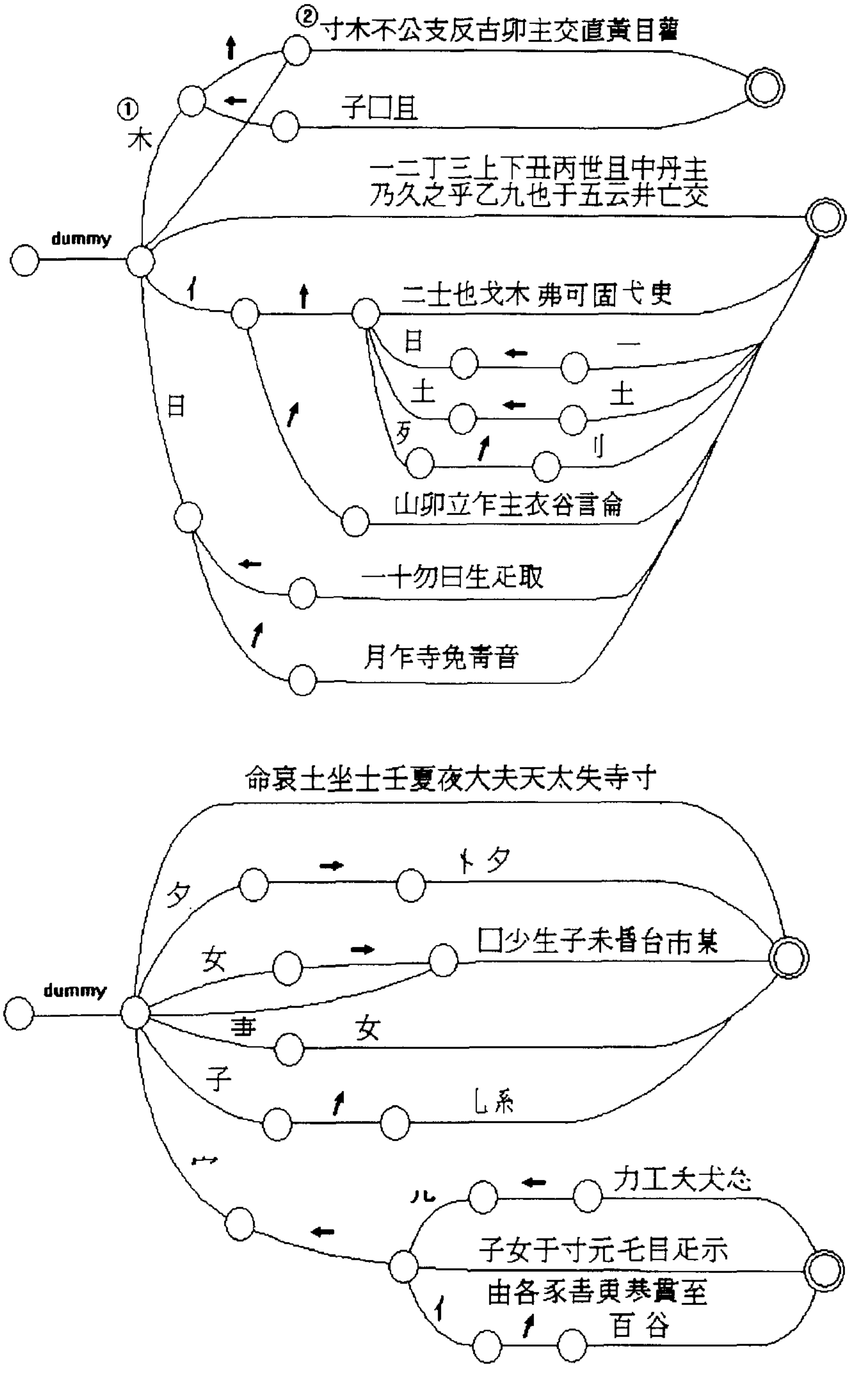


Fig. 3. State transition diagram for on-line Chinese characters.

edges means that different characters share the same HMM: for instance, the characters "木" on the edge ① and ② are shared by the same HMM for "木".

Each character on the edges is represented by using a radical HMM and each arrow is represented by using a ligature HMM as shown in Fig. 1. The dummy HMM on the leftmost edge is used to implement incorporation of these networks into a level building algorithm in Section 4.3. A path from the initial state to the final state depicts a complete character. Therefore, the large character set can be represented by a small number of HMMs.

The model still requires a lot of computing time to find an optimal path by evaluating probabilities for all the paths. We solve this problem with a level building algorithm in the following section. The level building algorithm keeps track of the path yielding maximum probability, which is generated by matching the unknown input pattern against a single HMM using the Viterbi algorithm.

4. RECOGNITION

4.1. Viterbi algorithm

A technique for finding the single best state sequence is the Viterbi algorithm[9] which is one of dynamic programming methods based on the principle of optimality. This method is so closely interrelated to the assumption of the Markov model and uses time constraints with the model that it can be applied to find an optimal path in HMMs. The Viterbi algorithm for a single HMM is as follows:

(1) *Initialization*:

$$\delta_1(i) = \pi_i b_i(O_1), \quad 1 \le i \le N, \tag{4}$$

$$\psi(i) = 0, \tag{5}$$

where N is the number of the state and $\delta_1(i)$ is the probability that symbol O_1 is observed at time $t=1$ and state i. The other variable ψ stores the optimal state at each time.

(2) *Recursion*:

$$\delta_t(j) = \max_{1<i\le N} [\delta_{t-1}(i)a_{ij}]b_j(O_t), \quad 2 \le t \le T, \ 1 \le j \le N, \tag{6}$$

$$\psi_t(j) = \arg\max_{1<i<N} [\delta_{t-1}(i)a_{ij}], \quad 2 \le t \le T, \ 1 \le j \le N, \tag{7}$$

where $\delta_t(j)$ is the maximum probability that the symbol $O_2, O_3, \ldots, O_T$ is observed at state j and time t. $\psi_t(j)$ stores the argument which maximizes $\delta_t(j)$ for each t and j to keep track of the state sequence.

(3) *Termination*:

$$p^* = \max_{1<i\le N} [\delta_T(i)], \tag{8}$$

$$q_T^* = \arg\max_{1\le i\le N} [\delta_T(i)]. \tag{9}$$

The maximum probability of $\delta_T(i)$ for each state i and at the final time is the scored probability of the observation sequence against the model, and the corresponding state is the final state of the optimal path. In the next step, backtracking happens from the final state to the initial state following the variable ψ_t.

(4) *State sequence backtracking*:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \ldots, 1. \tag{10}$$

The above expressions for the termination and backtracking for a single model case are modified in the level building algorithm described in the following section.

4.2. The level building algorithm

Given the individual radical models and ligature models W_j in the Chinese character recognition network, and given a test sequence of observation O_t, $t = 1, 2, \ldots, T$, corresponding to a sequence of eight-direction codes, recognition is to decode O into the sequence of models $W_{[1]}, W_{[2]}, \ldots, W_{[p]}$. Namely, it is to match the observation sequence to a state sequence of models with maximum joint probability.

To understand how we use the level building algorithm to find a sequence of models producing maximum probability, consider the case shown in Fig. 4, where we assume each model has five states.[17–19] We impose no constraints on the continuity of the model, that is, any model can follow any other model. At level l=1 (the initial level), we begin by matching a model $q(W_q)$ to observation sequence O, beginning at frame 1. To make the match we use a modified Viterbi decoding with the following form.

Initialization:

$$\delta_1(1) = [b_1^q(O_1)], \tag{11}$$

where $\delta_t(j)$ signifies the joint probability of partial state and observation sequences, $\Pr[O_1, O_2, \ldots, O_t$ and $S_1, S_2, \ldots, S_{t-1}, j \,|\, A, B]$, where S_i is the state at time t=i, and

$$\delta_1(j) = 0, \quad j = 2, 3, \ldots, N. \tag{12}$$

Recursion:

$$\delta_t(j) = \max_{1\le i\le N} [\delta_{t-1}(i) \times [a_{ij}^q]] \times [b_j^q(O_t)], \quad 2 \le t \le T, \ 1 \le j \le T. \tag{13}$$

Termination:

$$P(l, t, q) = \delta_t(N), \quad 1 \le t \le T. \tag{14}$$

$$B(l, t, q) = 0. \tag{15}$$

At the end of the level (when all models have been used), we reduce the level to form the arrays:

$$\hat{P}(l, t) = \max_q [P(l, t, q)], \tag{16}$$

$$\hat{B}(l, t) = B\left[l, t, \arg\max_q P(l, t, q)\right], \tag{17}$$

$$\hat{W}(l, t) = \arg\max_q [P(l, t, q)], \tag{18}$$

where $\hat{P}$ is the level output best probability, $\hat{B}$ is the level output backpointer, and $\hat{W}$ is the level output radical or ligature indicator.

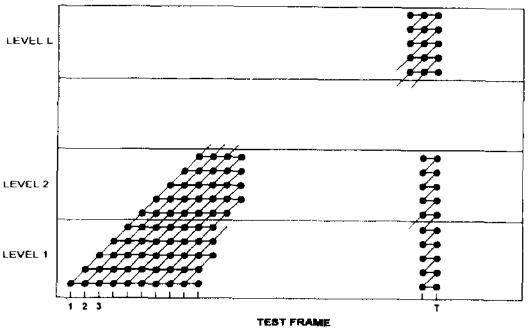


Fig. 4. Implementation of level building based on five-state HMMs.

The computation for level 2 (and all higher levels) differs only slightly in the initialization procedure. Since this level picks up from previous outputs, we have the initialization:

$$\delta_1(1) = 0, \quad \alpha_1(1) = 0, \tag{19}$$

$$\delta_1(1) = \max[\hat{P}(l-1, t-1), a^q_{11}\delta_{t-1}(1)] \times [b^q_1(O_t)], \quad 2 \le t \le T, \tag{20}$$

$$\alpha_t(1) = \begin{cases} t-1 & \text{if } \hat{P}(l-1, t-1) > \delta_{t-1}(1) \times a^q_{11}, \\ \alpha_{t-1}(1) & \text{otherwise.} \end{cases} \tag{21}$$

Equation (19) sets $\delta_1(1)$ to zero; equation (20) lets the level pick up at the most suitable place from the previous level. Equation (21) creates the appropriate initial backpointer array, which records the frame at the previous level in which the previous radical or ligature ended. During the recursion the backpointer is updated as

$$\alpha_t(j) = \alpha_{t-1}\left\{\arg\max_{1<i<N}[\delta_{t-1}(i) \times a^q_{ij}]\right\} \tag{22}$$

and at the end of the level, the probability and backpointer arrays become:

$$P(l, t, q) = \delta_t(N), \quad 1 \le t \le T, \tag{23}$$

$$B(l, t, q) = \alpha_t(N), \quad 1 \le t \le T. \tag{24}$$

Once all the models have been run at any level, the reduced $\hat{P}$, $\hat{B}$ and $\hat{W}$ arrays are formed and the computation proceeds to the next level.

The entire procedure terminates when some maximum number of level L is used, and in this point an adapted model sequence with probability $\hat{P}(L, T)$ is obtained by a backtracking algorithm using the backpointer array $\hat{B}(l, t)$. The backtracking algorithm is as follows:

```
Algorithm (Backtracking)
Begin
    level=L;
    back=T;
    while (level > 0) do
        begin
            $W_{[level]} = \hat{W}(level, back)$;
            $back = \hat{B}(level, back)$;
            $level = level - 1$;
        end
End
```

4.3. *Incorporation of construction rule into level building*

Modifying the above level building algorithm to handle the constraint network of the construction rule is straightforward. One merely makes the association that the levels are uniquely identified with states rather than radical or ligature position in complete characters, so that the candidates at lth level need not to be contiguous to those at the $(l+1)$st level. In this manner, only those models corresponding to radicals and ligatures leaving the lth state are matched. And another set of level backpointers is required to link the states (levels) in a temporal order.

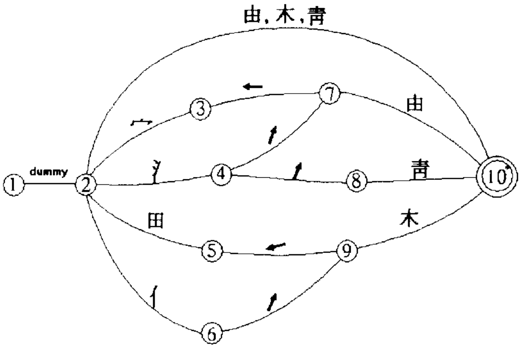


Fig. 5. Illustrative deterministic network for showing how construction rules can be incorporated into the level building.

Each level outputs a model generating a maximum probability of radical HMMs and ligature HMMs given time t. By the backtracking procedure for searching for an appropriate sequence of radical HMMs, a Chinese character is recognized at the end of the steps. The level building to handle the constraint does not implement the Viterbi decoding for all models in the network, but only implements it for models restricted for each level.

To illustrate the incorporation of construction rule into the level building, Fig. 5 shows a simple network for seven radicals and three ligatures of eight characters.

The final state in the network is indicated by an asterisk. Typical characters generated by the construction rule include the following characters: "由", "木", "青", "宙", "油", "清", "果" and "休".

The correspondence between states in the construction rule and levels in the level building algorithm can be seen in Table 1.

At level 1, we use the above modified Viterbi algorithm for each HMM to store a maximum probability for each frame (observation $O = O_1, O_2, \ldots, O_T$) in $\hat{P}(1, T)$ and store the index of that model producing a maximum probability in $\hat{W}(L, T)$. At this level, no model is indicated by backpointer (which at later levels is used to make the initial value of Viterbi decoding for the next level by deciding on a suitable model at each time t). Comparing the maximum probability generated at level 1 with the probability staying at the first state of HMMs in level 2, we run the initialization procedure by equations (20) and (21). After initialization, doing the recursion procedure of the Viterbi algorithm, we make a backpointer array by recording the maximum probabilities for each state into frames. This process continues recursively until the end of the maximum level. After doing all the processes, we continue the backtracking procedure to find an adaptive model sequence.

Table 1. The correspondence between states and levels

Current state	Radicals and ligatures used	Predecessor state	Current level	Predecessor levels	Index of model
2	Dummy model	1	0	0	0
3	山	2	2	0	4
4	氵	2	3	0	5
5	田	2	4	0	6
6	亻	2	5	0	7
7	Ligature ←	3	6	2	8
7	ligature ↗	4	7	3	9
8	ligature ↗	4	8	3	9
9	ligature ←	5	9	4	10
9	ligature ↗	6	10	5	9
10*	田, 青, 木	2	1	0	1, 2, 3
10*	田	7	11	6, 7	1
10*	青	8	12	8	3
10*	木	9	13	9, 10	2

This level building matches to the radical and ligature, one state at a time, based on a finite state transition diagram of the language. We select the Viterbi optimal path among the set of characters corresponding to terminal states (levels). For each terminal state (level) of the construction rule (see Fig. 5), the optimum path is traced back. The recognized character is the unique radical sequence corresponding to the most probable state sequence.

The following table shows an example of the detailed values of maximum probabilities, matching models, backpointers and previous level index during the recognition process. When the character "果" is given as observation symbols, the recognition path is from state ① to state ⑩ through states ②, ⑤ and ⑨. As a result of the Viterbi decoding we choose a radical "木" producing a maximum probability 1.598e−019 at the final level. Then we move to level 9 selected by the previous level index and choose a ligature model "←" indicated by the backpointer in level 13 at t=60. With the same method we move to level 4 and choose a radical model "田". Finally, the dummy does not need to be backtracked any more because it is useless for constructing a complete character.

T=50
Level: 0
$\hat{P}(1,t)$

$0.000e-000$	$1.563e-002$	$3.906e-003$	$9.766e-004$	$2.441e-004$	$6.104e-005$
$5.960e-008$	$1.490e-008$	$3.725e-009$	$9.313e-010$	$2.328e-010$	$5.821e-011$
$5.684e-014$	$1.421e-014$	$3.553e-015$	$8.882e-016$	$2.220e-016$	$5.551e-017$
$5.421e-020$	$1.355e-020$	$3.388e-021$	$8.470e-022$	$2.118e-022$	$5.294e-023$
$5.170e-026$	$1.292e-026$	$3.231e-027$	$8.078e-028$	$2.019e-028$	$5.049e-029$

$1.526e-005$	$3.815e-006$	$9.537e-007$	$2.384e-007$
$1.455e-011$	$3.638e-012$	$9.095e-013$	$2.274e-013$
$1.388e-017$	$3.469e-018$	$8.674e-019$	$2.168e-019$
$1.323e-023$	$3.309e-024$	$8.272e-025$	$2.068e-025$
$1.262e-029$	$3.155e-030$	$7.889e-031$	$1.972e-031$

$\hat{B}(1,t)$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$\hat{W}(1,t)$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Previous level

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Level: 4

$\hat{P}(4,t)$

0.000e − 000	0.000e − 000	0.000e − 000	0.000e − 000	0.000e − 000	0.000e − 000
9.352e − 040	1.957e − 028	9.886e − 025	1.330e − 021	1.188e − 021	1.466e − 020
3.002e − 014	1.733e − 012	1.698e − 011	2.285e − 008	2.156e − 005	4.304e − 005
1.963e − 015	3.918e − 015	7.819e − 015	1.561e − 014	3.115e − 014	5.883e − 017
1.429e − 026	1.897e − 026	1.897e − 029	2.321e − 031	1.385e − 031	2.611e − 031

0.000e − 000	0.000e − 000	3.463e − 317	5.787e − 093
1.260e − 010	2.515e − 010	5.020e − 010	7.530e − 013
2.152e − 008	4.065e − 011	7.677e − 014	1.450e − 016
1.111e − 019	8.333e − 023	6.250e − 026	5.272e − 027
2.540e − 029	1.270e − 032	6.349e − 036	1.111e − 038

$\hat{B}(4,t)$										$\hat{W}(4,t)$										Previous level									
0	0	0	1	1	3	4	5	1	1	6	6	6	6	6	6	6	6	6	6	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	6	6	6	6	6	6	6	6	6	6	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	6	6	6	6	6	6	6	6	6	6	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	6	6	6	6	6	6	6	6	6	6	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	6	6	6	6	6	6	6	6	6	6	0	0	0	0	0	0	0	0	0	0

Level: 9

$\hat{P}(9,t)$

0.000e − 000	0.000e − 000	0.000e − 000	0.000e − 000	0.000e − 000	0.000e − 000
0.000e − 000	7.263e − 100	2.208e − 043	7.509e − 029	3.793e − 025	5.103e − 022
9.213e − 015	3.531e − 017	6.644e − 017	6.650e − 013	1.732e − 015	7.170e − 016
1.349e − 011	6.746e − 015	3.373e − 018	1.686e − 021	2.453e − 022	1.843e − 018
1.793e − 026	2.689e − 029	3.146e − 030	1.041e − 028	1.861e − 031	3.256e − 034

0.000e − 000	0.000e − 000	0.000e − 000	0.000e − 000
2.552e − 025	4.599e − 028	3.954e − 018	1.973e − 017
1.015e − 012	7.622e − 009	1.434e − 008	2.698e − 008
1.195e − 014	1.195e − 017	1.195e − 020	1.195e − 023
1.628e − 037	1.229e − 038	1.793e − 036	2.689e − 039

$\hat{B}(9,t)$										$\hat{W}(9,t)$										Previous level									
0	0	0	0	0	0	0	0	0	0	10	10	10	10	10	10	10	10	10	10	4	4	4	4	4	4	4	4	4	4
8	9	10	11	12	13	13	15	16	17	10	10	10	10	10	10	10	10	10	10	4	4	4	4	4	4	4	4	4	4
18	19	15	21	22	23	24	25	25	25	10	10	10	10	10	10	10	10	10	10	4	4	4	4	4	4	4	4	4	4
25	25	25	25	32	33	34	34	34	34	10	10	10	10	10	10	10	10	10	10	4	4	4	4	4	4	4	4	4	4
34	34	34	41	41	41	41	45	46	46	10	10	10	10	10	10	10	10	10	10	4	4	4	4	4	4	4	4	4	4

Level: 13

$\hat{P}(13,t)$

0.000*e* − 000	0.000*e* − 000	0.000*e* − 000	0.000*e* − 000	0.000*e* − 000	0.000*e* − 000	0.000*e* − 000	0.000*e* − 000	0.000*e* − 000	0.000*e* − 000
0.000*e* − 000	0.000*e* − 000	0.000*e* − 000	2.458*e* − 164	9.715*e* − 066	2.205*e* − 049	5.175*e* − 040	4.145*e* − 034	2.314*e* − 034	2.892*e* − 037
2.169*e* − 040	1.084*e* − 043	1.084*e* − 046	1.084*e* − 049	4.071*e* − 047	1.252*e* − 041	5.572*e* − 038	5.572*e* − 041	5.572*e* − 044	5.572*e* − 047
7.355*e* − 043	4.105*e* − 043	3.355*e* − 042	5.791*e* − 039	3.232*e* − 039	3.232*e* − 042	3.232*e* − 045	5.218*e* − 043	4.902*e* − 037	2.477*e* − 035
1.411*e* − 036	7.612*e* − 036	2.490*e* − 030	1.154*e* − 029	1.053*e* − 026	1.293*e* − 020	5.151*e* − 017	7.409*e* − 017	1.066*e* − 016	1.598*e* − 019

$\hat{B}(13,t)$										$\hat{W}(13,t)$										Previous level									
0	0	0	0	0	0	0	0	6	7	2	2	2	2	2	2	2	2	2	2	0	0	9	9	9	9	9	9	10	10
7	7	7	6	7	7	7	7	7	7	2	2	2	2	2	2	2	2	2	2	10	10	10	10	10	10	10	4	10	10
7	7	7	7	14	15	15	15	15	15	2	2	2	2	2	2	2	2	2	2	9	9	9	9	9	9	9	9	9	9
20	20	23	23	23	23	23	28	28	29	2	2	2	2	2	2	2	2	2	2	9	9	9	9	9	9	9	9	9	9
29	29	29	29	29	29	29	29	29	29	2	2	2	2	2	2	2	2	2	2	9	9	9	9	9	9	9	9	10	9

5. EXPERIMENTAL RESULTS

We implemented the on-line cursive Chinese character recognizer on an MS-Windows NT platform of a Pentium 90 computer as shown in Fig. 6. The program was compiled using Borland Object Windows for C++ 2.0. To evaluate the practical performance of the proposed method, experiments were done for 10 sets of 1800 daily-used basic Chinese characters in Korea. The data were written by 10 different writers without any constraints on the WACOM SD-510C digitizer. Figure 7 shows sample test characters.

5.1. Data coding and training

An input stroke is a sequence of absolute (*x*,*y*) positions on a digitizer while writing. Since the sequence contains many variations, a preprocessing is needed to remove the variations. To correctly extract stroke features representing radicals and spaces occurring inside radicals or between radicals, a resampling process is needed. The input character can be converted into similar sized points regardless of the absolute size of the character if the resampling is processed as the ratio of the size of character. We resample 70 points from a character and generate a sequence of eight-direction codes using a direction between sampled points.

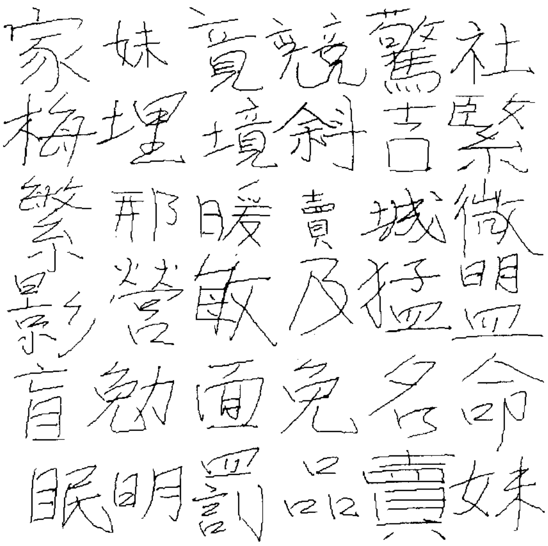
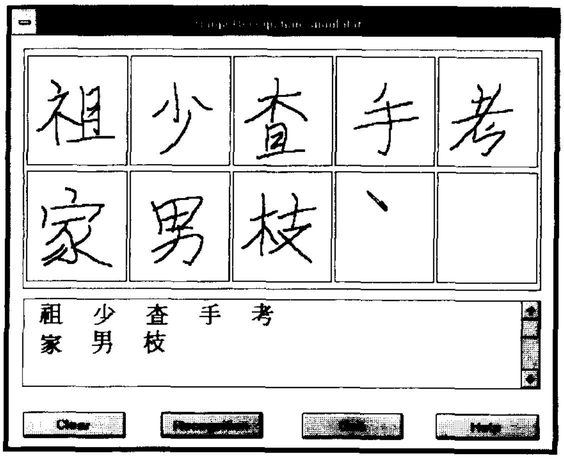


Fig. 7. Sample characters for test.

Then we train each single radical HMM and ligature HMM using the above observation sequences. The HMM training process needs a lot of sample data to make a better model because of many probabilistic parameters. Since the data do not represent radicals divided separately but represent a whole character, it is difficult for us to extract a boundary point from the data. So we have to



祖　少　查　手　考
家　男　枝

Fig. 6. Screen layout.

Table 2. Recognition rate

Source	No. of characters	Misrecognition		Reject		Correct recognition	
		#	%	#	%	#	%
Trained character	7200	403	5.6	0	0.0	6707	94.4
Test character	10,800	822	7.6	226	2.1	9752	90.3

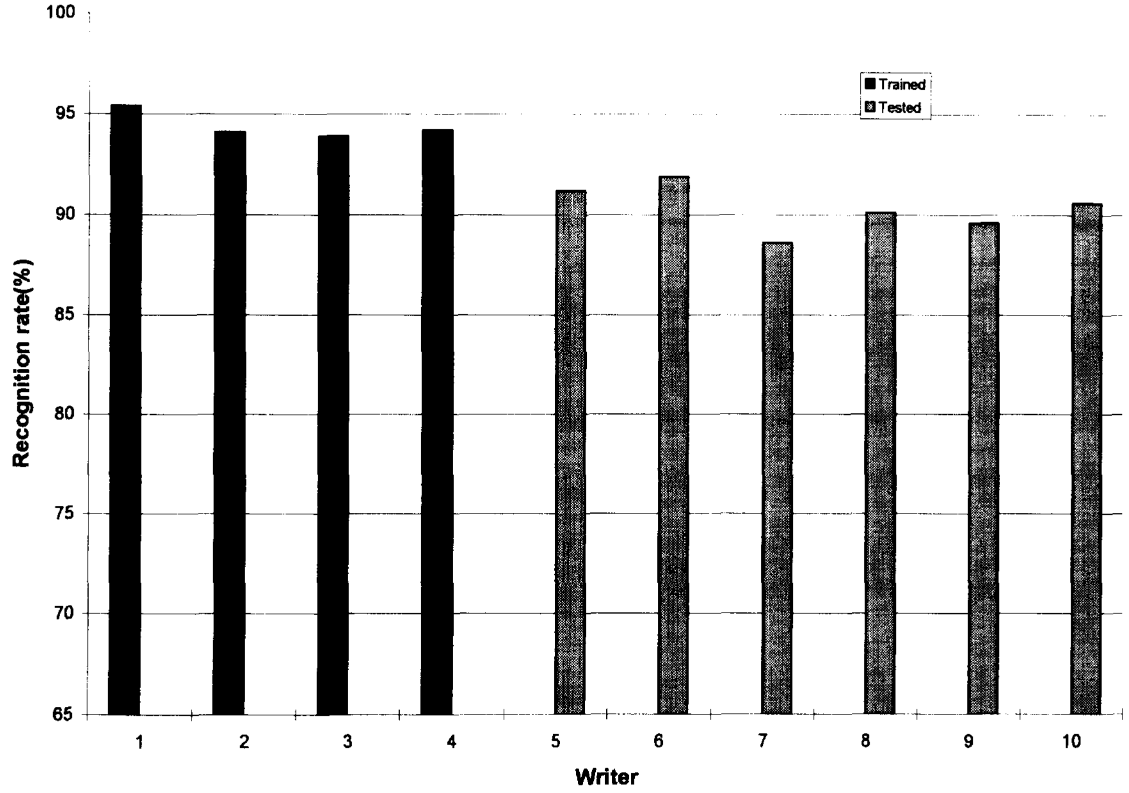


Fig. 8. Recognition rate versus test character set.

indicate the boundary point between radicals manually. This boundary point must be chosen to be appropriate for a basic Chinese character form.

5.2. *Chinese character recognition*

Data for recognition are taken by the same method as in the training procedure. In the network, level building recognizes a Chinese character using these data. Table 2 shows the recognition rate of the experiment. The correct recognition rates for trained characters and test characters were 94.4% and 90.3%, respectively. Figure 8 shows the recognition rate of each character set. It shows consistent recognition rates for the ten character sets. This result reflects that HMM has superior capabilities for extracting and modeling knowledge from sources.

The main recognition errors were from abnormal writing and ambiguity. Figure 9 shows example characters that were misrecognized. The character in (a) is rejected because the writing order of the first stroke and the second stroke is changed incorrectly. The abnormal writing occurred frequently. This problem can be simply solved by expanding the recognition network as adding HMM models representing incorrect writing order. The confusion set of characters caused by an ambiguous shape is shown (b). For instance, the data sampled by eight-direction codes of "人", "入" and "八" are as follows:

Character	Sampling data
人	2 2 2 2 2 2 2 3 3 3 3 3 4 4 7 7 7 7 7 7 7 7 1 0 1 1 1 1 1 1
入	2 2 2 2 3 3 3 4 5 7 7 7 7 7 7 7 7 7 1 1 1 1 1 1 1 1 1 1 1 1
八	2 2 2 3 3 3 3 4 7 7 7 7 7 7 7 7 7 7 7 7 1 1 1 1 1 1 1 1 1 1

Because the traces of strokes in each character are almost similar, it is very difficult to identify these characters using HMMs. So we need an additional method that tries to tell apart similar Chinese characters by utilizing the heuristic knowledge. Table 3 shows the relative error ratio according to these error types.

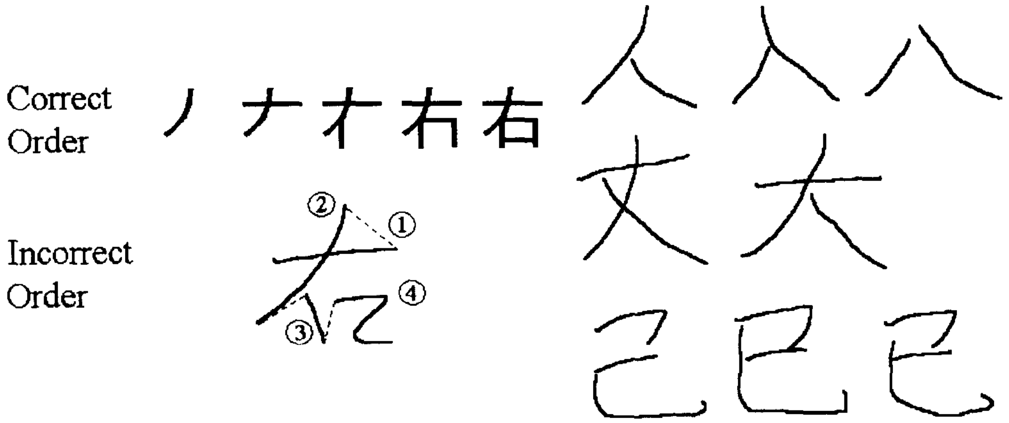


Fig. 9. Examples of recognition error.

Table 3. Error ratios according to error types

Error type	Number of error characters	Rate (%)
Abnormal writing	258	24.6
Ambiguity	790	75.4
Total	1048	100.0

6. CONCLUSIONS

In this paper, we have presented a stochastic network model for on-line recognition of handwritten Chinese characters, having HMM as a main statistical framework. The network model is composed of radical and ligature HMMs supposing that a Chinese character is a set of consecutive and meaningful radicals. As a search strategy, it uses a modified level building algorithm. The search algorithm has a time complexity that depends only on the number of levels because it determines a radical or a ligature in the unit of level while keeping track of the path yielding maximum probability for an input pattern.

In the experiment, the proposed model permitted a large degree of flexibility in dealing with shape and size variations and represented the larger character set efficiently with a small number of HMMs. It also showed an excellent recognition speed. These results suggest that it is reasonable and feasible for handwritten Chinese characters. Though the performance of the proposed model is very good, it could be improved further in at least three respects: (1) through the study of writing behavior, the characters written abnormally could be accommodated; (2) an algorithm to tell apart the similar characters and the recognition using candidate generated by a level building would improve the recognition rate; (3) above all, more training data would lead to a more practical recognition system. Future work could also include extracting more robust features for HMMs to achieve better discrimination power.

REFERENCES

1. C. Tappert, C. Y. Suen and T. Wakahana, The state of the art in on-line handwriting recognition, *IEEE Trans. Pattern Analysis Mach. Intell.* **12**(8), 787–808 (1990).
2. T. Wakahana, H. Murase and K. Odaka, On-line handwriting recognition, *Proc. IEEE* **80**(7), 1181–1194 (1992).
3. K. C. Fan, C. K. Lin and K. S. Chou, Confusion set recognition of on-line Chinese characters by artificial intelligence technique, *Pattern Recognition* **28**(3), 303–313 (1995).
4. W. Y. Chen, Y. F. Liao and S. H. Chen, Speech recognition with hierarchical recurrent neural networks, *Pattern Recognition* **28**(6), 795–805 (1995).
5. S. I. Hanaki and T. Yamazaki, On-line recognition of handprinted Kanji characters, *Pattern Recognition* **12**, 421–429 (1980).
6. A. J. Hsieh, K. C. Fan and T. I. Fan, Bipartite weighted matching for on-line handwritten Chinese character recognition, *Pattern Recognition* **28**(2), 143–151 (1995).
7. C. K. Lin, K. C. Fan and T. P. Lee, On-line recognition by deviation-expansion model and dynamic programming matching, *Pattern Recognition* **26**(2), 259–268 (1993).
8. W. T. Chen and T. R. Chou, A hierarchical deformation model for on-line cursive script recognition, *Pattern Recognition* **27**(2), 205–219 (1994).
9. L. R. Rabiner, Tutorial on hidden Markov model and selected application in speech recognition, *Proc. IEEE* **77**(2), 257–285 (1989).
10. L. R. Rabiner, S. E. Levinson and M. M. Sondhi, On the application of vector quantization and hidden Markov models to speaker-independent, isolated word recognition, *Bell System Tech. J.* **62**(4), 1075–1105 (1983).
11. L. R. Rabiner, S. E. Levinson and M. M. Sondhi, An introduction to the application of the application of the theory of probabilistic function of a Markov process to automatic speech recognition, *Bell System Tech. J.* **62**(4), 1035–1074 (1983).
12. Bose B. Chinmoy and Shyh-Shiaw Kuo, Connected and degraded text recognition using hidden Markov model, *Pattern Recognition* **27**(10), 1345–1363 (1994).
13. J. W. Kim, B. K. Sin and J. H. Kim, Training HMM by giving initial parameter estimates: An application to on-line character recognition, *The First Character Recognition Workshop*, pp. 163–168 (1993).
14. S. B. Cho and J. H. Kim, Integrating hidden Markov model and neural network classifier for on-line handwritten character recognition, *J. KISS* **20**(3), 328–337 (1993).
15. B. K. Shin and J. H. Kim, On-line cursive Hangul character recognition using a hidden Markov model network, *J. KISS* **21**(9), 1737–1745 (1994).
16. A. Kundu, Y. He and P. Bahl, Recognition of handwritten word: First- and second-order hidden Markov model based approach, *Pattern Recognition* **22**(3), 283–297 (1989).

17. Cory S. Myers and L. R. Rabiner, A level building dynamic time warping algorithm for connected word recognition, *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-29**(2), 284–297 (1981).
18. Cory S. Myers and S. E. Levinson, Speaker independent connected word recognition using a syntax-directed dynamic programming procedure, *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-30**(4), 561–565 (1982).
19. L. R. Rabiner and S. E. Levinson, A speaker-independent, syntax-directed, connected word recognition system based on hidden Markov models and level building, *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-33**(3), 561–573 (1985).
20. J. H. Bae, J. Y. Yang, P. K. Kim and H. J. Kim, On-line handwritten Hangul character recognition including unstandardized pattern, in *Proc. 20th KISS Spring Conf.*, pp. 159–162 (1993).
21. S. B. Cho and J. H. Kim, Integrating hidden Markov model and neural network classifier for on-line handwritten character recognition, *J. KISS* **20**(3), 328–337 (1993).
22. P. K. Kim, J. Y. Yang and H. J. Kim, On-line cursive Korean character recognition using extended primitive strokes, in *Proc. Third PRICAI* (*Pacific Rim Int. Conf. on A.I.*), pp. 816–821 (1994).
23. J. W. Kim, K. C. Jung, S. K. Kim and H. J. Kim, Shape classification of on-line Chinese character strokes using ART-1 neural network, in *Proc. World Congress on Neural Networks* (*WCNN'95*) **I**, 54–57 (1995).
24. S. K. Kim, J. W. Jung, J. W. Kim and H. J. Kim, On-line recognition of Chinese characters based on ART-1 neural network, *J. Korea Institute of Telematics and Electronics* (*KITE*) **33**(2), 368–377 (1996).

About the Author—HANG JOON KIM received the B.S. degree in Electrical Engineering from the Seoul National University, Seoul, S. Korea, in 1977 and the M.S. degree in Electrical and Electronic Engineering from the Korea Advanced Institute of Science and Technology in 1979. From 1979 to 1983, he was a full-time Lecturer at the Department of Computer Engineering, Kyungpook National University, Taegu, S. Korea, and from 1983 to 1994 an Assistant and Associate Professor at the same department. Since October 1994, he has been with the Kyungpook National University as a Professor. His research interests include image processing, pattern recognition and artificial intelligence.

About the Author—KYUNG HYUN KIM received the B.S. degree in Statistics from the Kyungpook National University, Taegu, S. Korea in 1995. He will receive the M.S. degree in Computer Engineering from Kyungpook National University. Major fields of interest include pattern recognition and neural networks.

About the Author—SANG KYOON KIM received the B.S. degree in Statistics from Kyungpook National University, Taegu, S. Korea, in 1991 and the M.S. and Ph.D. degrees in Computer Engineering from Kyungpook National University, Taegu, S. Korea, in 1994 and 1996, respectively. Since September 1996, he has been with the Department of Computer Science, Inje University, Kimhae, S. Korea. His research interests include pattern recognition and artificial intelligence.

About the Author—JONG KOOK LEE received the B.S. degree in Statistics from Kyungpook National University, Taegu, S. Korea, in 1989 and the M.S. and Ph.D. degrees in Computer Engineering from Kyungpook National University, Taegu, S. Korea, in 1991 and 1995, respectively. From 1993 to 1996, he was a full-time instructor in the Department of Computer Engineering, Dongseo University, Pusan, S. Korea. In March 1996, he joined the faculty of the Department of Computer Engineering, Andong National University, Andong, S. Korea. His research interests include pattern recognition, image processing and neural networks.