pISSN 1976-913X
eISSN 2092-805X

J Inf Process Syst, Vol.8, No.4, December 2012 http://dx.doi.org/10.3745/JIPS.2012.8.4.653

Online Recognition of Handwritten Korean and English Characters

Ming Ma*, Dong-Won Park*, Soo Kyun Kim* and Syungog An*

Abstract—In this study, an improved HMM based recognition model is proposed for online English and Korean handwritten characters. The pattern elements of the handwriting model are sub character strokes and ligatures. To deal with the problem of handwriting style variations, a modified Hierarchical Clustering approach is introduced to partition different writing styles into several classes. For each of the English letters and each primitive grapheme in Korean characters, one HMM that models the temporal and spatial variability of the handwriting is constructed based on each class. Then the HMMs of Korean graphemes are concatenated to form the Korean character models. The recognition of handwritten characters is implemented by a modified level building algorithm, which incorporates the Korean character combination rules within the efficient network search procedure. Due to the limitation of the HMM based method, a post-processing procedure that takes the global and structural features into account is proposed. Experiments showed that the proposed recognition system achieved a high writer independent recognition rate on unconstrained samples of both English and Korean characters. The comparison with other schemes of HMM-based recognition was also performed to evaluate the system

Keywords—Online Handwriting Recognition, Hidden Markov Model, Stochastic Grammar, Hierarchical Clustering, Position Verifier

1. INTRODUCTION

Even though online handwriting recognition has been researched for over four decades; it is still a tough problem. Due to the recent advances achieved in hardware technology and the emergence and growing popularity of handheld devices, such as Personal Digital Assistants (PDAs), mobile phones, and Ultra Mobile PCs (UMPCs), new methods for input, besides the keyboard and mouse, such as recognizing speech or handwriting, have been developed. People without prior training can easily learn to use them because of this natural means of communication. This development inspires new applications of handwriting recognition and has led to some novel interests in research.

Unfortunately, neither of the recognition problems has been completely solved yet. The most prominent problem in handwriting recognition is the vast variation in personal writing styles. There are also a lot of variations within the writing style of one person. These variations depend on things like the context of the writing, the writing equipment, the writing situation, and the

Manuscript received May 15, 2012; accepted May 21, 2012.
Corresponding Author: Dong-Won Park
* Dept. of Game Engineering, Paichai University, Daejeon, South Korea ({ming, dwpark, kimsk, sungohk}@pcu.ac.kr)

mood of the writer. The writing style may also evolve with time or practice. Thus, the performance of the automatic recognition system depends heavily on how well the different personal writing styles and their variations are modeled. [15]

Another research problem in this project concerns Korean character processing. Unlike English, the Korean language has 51 graphemes and a large set of characters. Thus, recognizing a given Korean character can be different from, or even more difficult than, an English word. A Korean character is composed of two or three graphemes that are arranged in two dimensions and not in a simple left-to-right concatenation.

2. OVERVIEW OF THE ONLINE HANDWRITING RECOGNITION SYSTEM

The main goal of this study is to develop a practical online recognition system for handwritten English and Korean characters. The system should be able to handle multiple writing styles and cursive forms of handwritten characters. The HMM-based recognition model is well suited for this requirement as has been verified in many research studies. In general, our system can be divided into two parts – the training part and the recognition part (Fig. 1). Though the two parts are functionally separate from each other, both the training and recognition parts share some components like preprocessing and feature extraction.

The training part takes a batch of training data as input. After preprocessing and feature extraction, the extracted feature vectors come into the clustering component. Various handwriting styles in a class are grouped and trained into different models through the training engine. The models are then stored in the model database for recognition.

The recognition occurs by inputting the raw data into the recognition part. The same preprocessing and feature extraction process, as is done in the training part, are performed for the raw data. The recognition engine is activated when a feature vector is presented. The final output, which is a ranked list of recognition candidates, is displayed on the screen.

The preprocessing of the trajectory of the input pattern directly facilitates pattern description and affects the quality of the description [9]. In this project, two sampling approaches – equidistant data points and corner detection – are used and compared to find a better solution.

Equidistant data points: the trajectory points are resampled in a way where the distance between adjacent points is approximately equal. The amount of data in the equidistant data point

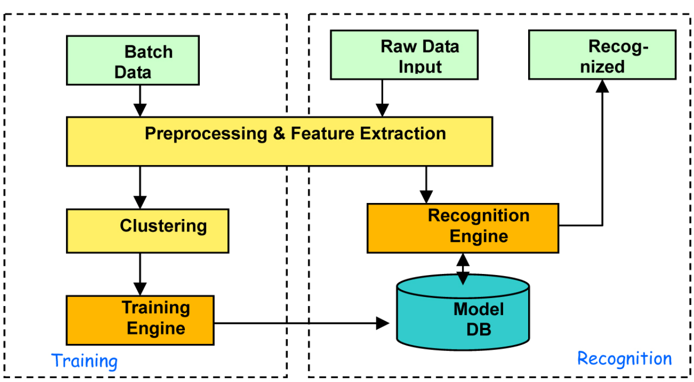


Fig. 1. General overview of the developed recognition system

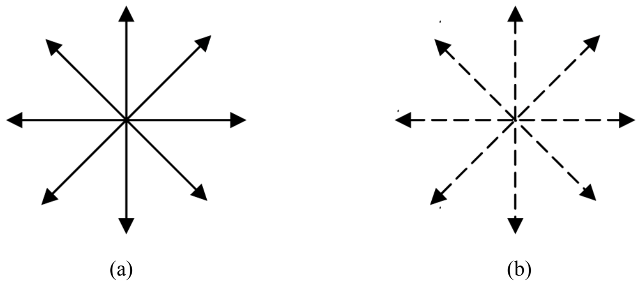


Fig. 2. (a) Pen-down codes, (b) Pen-up codes

representation is still appreciable.

Corner detection: the trajectory points are only kept when the direction of those data points changes significantly. A higher data reduction rate can be achieved by detecting the corner points of the trajectories. (Fig. 2)

In the process of feature extraction, the tangent direction of the pen-down and pen-up trace is encoded by one of 8-direction codes, as shown in Fig. 2.

3. MODEL OF HANDWRITING

This section is concerned with how to design 'hidden Markov models' (HMMs) for letters, ligatures, and words. We have omitted the description about the HMM, as it is mentioned in Section 3.5

3.1 Modeling Unit

In the English language, letters are the most basic and natural units for describing words. For each letter, a six-state HMM is designed. While in the Korean language, graphemes are the basic units to construct words. For each grapheme, we designed a HMM of various states according to the shape complexity of the grapheme. In a Markov chain, the number of states is a measure, albeit crude, of the complexity of the finite state grammar represented by that chain. Each state represents an event or a local characteristic pattern in a signal [1]. Likelihood further improves as N increases to some extent and the error rate seems to reach a minimum at a specific value of N [13] through the model. Therefore, as showed in Fig. 3, we decide the number of states to be small and proportional to the shape complexity of the grapheme.

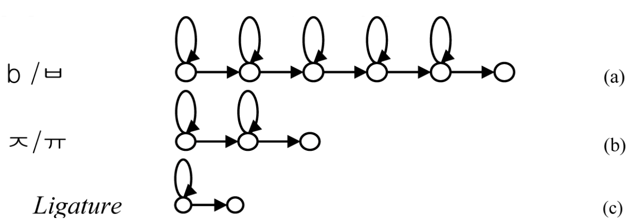


Fig. 3. Various HMM states that are based on the shape complexity

Each HMM is designed with the type of left-to-right transition topology that captures the temporal characteristic of online handwriting signals.

3.2 Ligature Model

The imaginary lines between strokes are referred to as “ligatures,” which are modeled explicitly through the pen-up traces. A two-state HMM is designed for ligatures between two graphemes. (Fig. 3(c))

3.3 Korean Character Model

There are 24 primitive graphemes in Korean characters. The graphemes can be further classified as consonants and vowels. Ten of the primitive graphemes are vowels and the rest are consonants. Complex vowels and consonants are made by combining simple vowels and simple consonants, respectively [7]. In addition, the primitive graphemes of consonants construct another 16 double consonants, whereas the primitive graphemes of vowels construct another 6 complex vowels. In this study, we proposed a novel Korean character model based on a modified level building network. It consists of a series of grapheme models that are embedded in a structured network according to rigorous composition syntax, which is composed of grapheme order constraint and structure constraint [7].

3.4 HMMs Network

Once the grapheme and the ligature models have been established, a 5-layer finite state network (FSN) called BongNet, which was designed by [7], is used as the baseline HMMs network, as it effectively represents the architecture of the Korean characters. As we can see from Fig. 4, a set of dummy nodes (initial and final nodes) is placed at each end of the network. Each path from the initial node to the final node corresponds to a Korean character. Each node between the first consonant and the ligature corresponds to a group of C graphemes with similar ending strokes. The nodes between first ligature and vowel correspond to groups of V graphemes with similar starting strokes. A second and fourth layer represents all of the 40 ligature models.

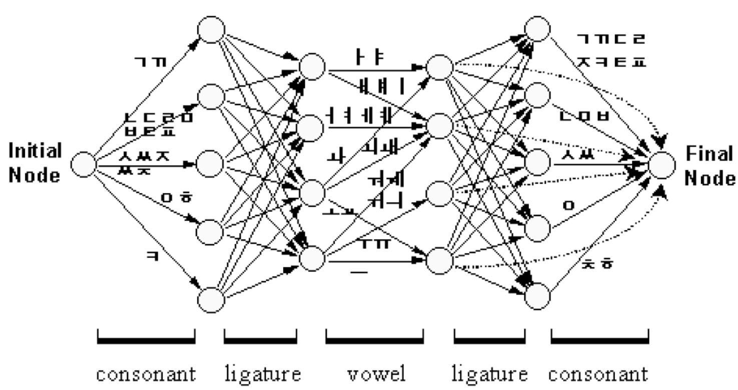


Fig. 4. HMMs network - *BongNet*

3.5 Clustering for The Multiple Models Design

Due to the fact that handwritten characters can vary in both writing order and general shape (Fig. 5), using only one model for all patterns of a letter or grapheme may weaken the recognition performance of the system. Thus, several models are needed to represent the letters or graphemes with different writing orders or shapes. To this end, a clustering technique is usually applied to obtain multiple models for a single letter or grapheme. The difficulties then lie in how to decide on the number of models in advance for top-down clustering, and similarly, how to select a proper distance measure for clustering samples without any additional knowledge or constraints.

Fig. 5. Different writing style of the letter "b" and the Korean grapheme "ㅂ"

3.6 The Design for Multiple Hmms

After introducing the clustering into a model design, the next question is how to effectively arrange the models of the same letters or graphemes during matching. Due to the existence of there being over two models for a letter or grapheme, models of the same letter or grapheme may compete for selection, which will impair the performance of recognition. Therefore, to combine the multiple models of the same letter or grapheme into a single HMM, a multiple parallel-path of HMMs is constructed as shown in Fig. 6.

A set of dummy nodes (initial and final nodes) is placed at both ends of the models of the same letters or graphemes. All of these models are arranged in parallel and connect to dummy nodes directly. There is no connection between models. Thus, each model contributes to one of the multiple paths from the initial node to the final node. Refer to Fig. 6(a).

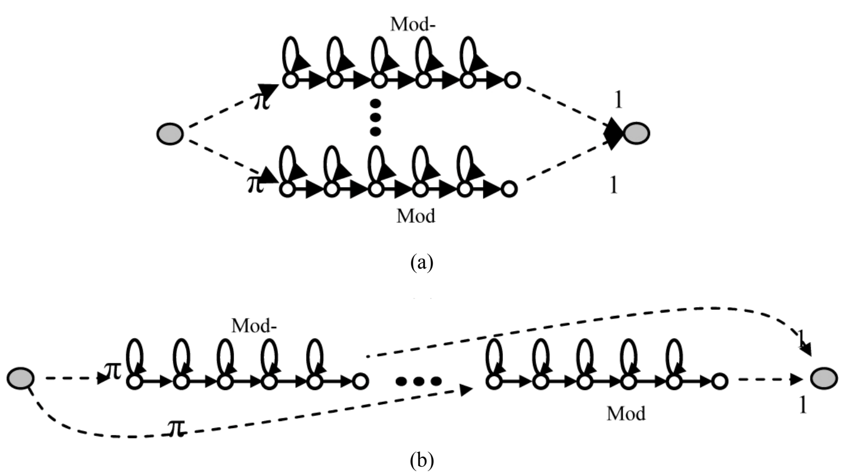


Fig. 6. Architecture of multiple parallel-path HMMs

3.7 Training of HMMs

Models are trained using the well-known Baum-Welch re-estimation algorithm. Training is first conducted on the isolated letter samples and on manually segmented graphemes and ligatures from Korean characters. Each model is trained with the data only in the corresponding cluster. Then multiple models with the same class label are trained for the multiple parallel-path HMMs.

In the training procedure, a new model λ*'* is created by re-estimating the parameters of a given model λ using the feature vectors of the training samples. The aim of the training is to find the model λ*'*, such that:

$$\lambda' = \arg\max_{\lambda} P(O \mid \lambda) \tag{1}$$

where the O is the given observation sequence and the $P(O \mid \lambda)$ is the likelihood of that sequence given the model λ. The procedure to find the model that maximizes likelihood is the so-called forward-backward algorithm.

The forward probability can be calculated by the following recursions:

$$\alpha_i(t) = [\sum_{j=1}^{N} a_j(t-1)a_{ji}]b_i(O_t) \tag{2}$$

where $\alpha_i(t)$ is defined as a forward probability, which is the joint probability of having generated the partial observation sequence from time 1 to time t and having arrived at state i at time t. N is the total number of states in the given HMM, and

$$\beta_i(t) = \sum_{j=1}^{N} a_{ij}b_j(O_{t+1})\beta_j(t+1) \tag{3}$$

where $\beta_i(t)$ is defined as a backward probability, which is the probability of generating the partial observation sequence from time to time T, given an HMM and that the state sequence starts from state i at time t.

Therefore, the product of these two $\alpha_i(t)\beta_i(t)$ denotes the joint probability of generating the incoming observation sequence and state i arriving at time t. Note that at any time t, all possible state sequences must merge into one of the states. Thus, the desired probability $P(O \mid \lambda)$ is simply computed by summing all of the forward and backward products as shown below:

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_i(t)\beta_i(t) \tag{4}$$

For an HMM λ with mixture components, the means, covariance matrices, mixture weights, and transition probabilities are re-estimated as follows:

$$\hat{u}_{im} = \frac{\sum_{t=1}^{T} \delta_{im}(t) O_t}{\sum_{t=1}^{T} \delta_{im}(t)} \tag{5}$$

$$\hat{c}_{im} = \frac{\sum_{t=1}^{T} \delta_{im}(t)(O_t - \hat{u}_{im})(O_t - \hat{u}_{im})'}{\sum_{t=1}^{T} \delta_{im}(t)} \tag{6}$$

$$w_{im} = \frac{\sum_{t=1}^{T} \delta_{im}(t)}{\sum_{t=1}^{T} \delta_{i}(t)} \tag{7}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T} \alpha_i(t) a_{ij} b_j(O_{t+1}) \beta_j(t+1)}{\sum_{t=1}^{T} \alpha_i(t) \beta_i(t)} \tag{8}$$

where $\delta_{im}(t)$ denotes the probability of the observation sequence occupying the m^{th} mixture component of state i at t time, and $\delta_i(t)$ denotes the probability of the observation sequence occupying the state i at time t. They can be expressed as follows:

$$\delta_i(t) = \sum_{m=1}^{M} \delta_{im}(t) = \sum_{m=1}^{M} \frac{1}{P} \sum_{j=1}^{N} \alpha_j(t-1) \alpha_{ji} w_{im} b_{im}(O_t) \beta_i(t) \tag{9}$$

where M is the total number of Gaussian mixture components in state I and N is the total number of states in the model.

By using these re-estimation formulas, we could easily extend it to handle the case where the incoming observation is constructed by multiple HMMs and the parameters are estimated by multiple iterations.

4. Recognition methods

4.1 Modified Level Building Algorithm

The level building algorithm [13] is used to match a series of HMMs to an observation sequence without having to first segment the sequence into sub-sequences that was produced by different models. Each level of the level building algorithm corresponds to match a character model to some part of the observation sequence. In our system, a proposed modified level building algorithm is used as illustrated in Fig. 7.

If we denote that the set of V models of the graphemes and ligatures as λ^v, $1 \le v \le V$, in

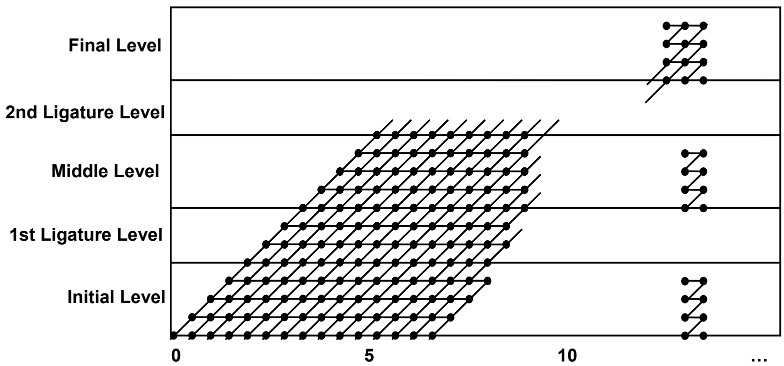


Fig. 7. Modified level building based on HMMs of varied states

the Korean character recognition network, and a test sequence of observation O_t, $t = 1, 2, \ldots, T$, corresponding to a feature vector, which is a sequence of 16-direction codes, then the recognition is to decode O into the sequence of models $\{\lambda^1, \lambda^2, \ldots, \lambda^l\}$. Namely, it is to match the observation sequence to a state sequence of models with maximum joint probability.

Fig. 7 illustrates how HMMs are applied in the level building algorithm. Five levels representing each component of a Korean character are explicitly modeled in the level building algorithm. The number of states in each grapheme level is varied according to the models of grapheme. The constraints of grapheme order and structure that we introduced for the Korean character model are imposed at the end of each level. For each HMM λ^v, and at each level l, we do a Viterbi match against O, starting from frame 1 on level 1. A complete matching procedure is described as follows: [5]

Initialization:

$$\delta_1(1) = [b_1^q(O_1)], \tag{10}$$

where $\delta_1(j)$ 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

$$\delta_1(j) = 0, \quad j = 2,3,\ldots,N \tag{11}$$

Recursion:

$$\delta_t(j) = \max_{1 \le i \le N}[\delta_{t-1}(i) \times [a_{ij}^q]] \times [b_j^q(O_t)] \qquad 2 \le t \le T, \; 1 \le j \le T \tag{12}$$

Termination:

$$P(l,t,q) = \delta_t(N), \quad 1 \le t \le T$$
$$B(l,t,q) = 0$$

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

$$\hat{P}(l,t) = \max_{q}[P(l,t,q)] \tag{13}$$

$$\hat{B}(l,t) = B\left[l,t,\arg\max_{q} P(l,t,q) \right] \tag{14}$$

$$\hat{W}(l,t) = \arg\max_{q}[P(l,t,q)] \tag{15}$$

where $\hat{P}$ records the best output level of accumulated log to t at level l, $\hat{B}$ records the back pointer of the best model, and $\hat{W}$ records the label of the grapheme or ligature that was output at level l. The computation for level 2 (and all higher levels) differs only slightly in the initialization procedure. Since these levels pick up from previous outputs, we have the initialization:

$$\delta_1(1) = 0\,, \alpha_1(1) = 0$$
$$\delta_t(1) = \hat{P}(l-1,t-1),\;\; 2 \le t \le T$$

where $\alpha_t(j)$ is the initial back pointer array, which records the frame t in the end of the previous level. The $\delta_1(1)$ and $\alpha_1(1)$ is set to zero at the beginning of level 2 at a higher level. Then a starting frame is picked up from the preceding level based on $\hat{P}(l-1,t-1)$. $\alpha_t(j)$ and is initialized and updated through the following equations:

$$\alpha_t(1) = \begin{cases} t-1 & if\ \hat{P}(l-1,t-1) > \delta_{t-1}(1) \times a_{11}^{q} \\ \alpha_{t-1}(1) & otherwise \end{cases} \tag{16}$$
$$\alpha_t(j) = \alpha_{t-1}\left[\arg\max_{1\le i\le N} |\,(\delta_{t-1}(i) \times a_{11}^{q}) \right]$$

and at the end of the level, the probability and back pointer arrays become:

$$\begin{aligned} P(l,t,q) &= \delta_t(N), \quad 1 \le t \le T \\ B(l,t,q) &= \alpha_t(N), \quad 1 \le t \le T \end{aligned} \tag{17}$$

Once this process is repeated through all these five levels, an adapted model sequence with probability $\hat{P}(L,T)$ is obtained at this point by a backtracking algorithm using the back pointer array $\hat{B}(l,t)$.

4.2 Post-Processing Methods

Although the HMM based statistical method has many advantages, it still has some limitations that are caused by the assumption of missing global properties. Fig. 8 shows the typical confusion over the similar graphemes made by the general HMM based method. In these examples, each two graphemes are very similar to each other if a curtain stroke in the grapheme is omitted. In the recognition, since these strokes are nothing but small parts of graphemes, their

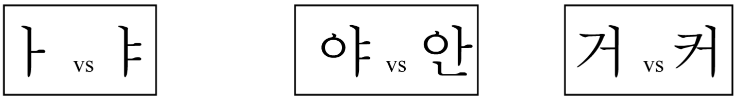


Fig. 8. Examples showing the limitation of the HMM based method

mismatching hardly influences the likelihood of HMM [7].

Therefore, to solve this problem, the global and structural knowledge about the characters is utilized in a post-processing procedure. Two kinds of the structural analyzers—the shape verifier and position verifier—are used in this study. It should be noted that the verifiers are only applied in recognition of a Korean character.

4.2.1 Shape verifier

Due to the inherent limitation of the Markov assumption, an HMM represents only a sequence of local properties. Because of this, the omission of a small important stroke may not cause enough likelihood for a change of HMM during recognition. The example showed in Fig. 9 is likely to be mismatched. Therefore, a stochastic grammar [7] is introduced as a shape verifier. To build the stochastic grammar, the representative graphemes in each cluster generated for multiple HMMs design are identified via a sequence of primitive strokes such as a vertical line or a horizontal bar. Stochastic grammar is characterized by the following elements:

1) ***N***: the number of representative shape states for a given grapheme.
2) ***M***: the number of primitive strokes for a given grapheme.
3) ***S***: the representative shape states vector. We denote it as $S = \{s_0, s_1, \ldots, s_N\}$.
4) ***V***: the terminal symbol vector encoded for primitive strokes.
We denote it as $V = \{v_0, v_1, \ldots, v_M, \varepsilon\}$, where ε is an unknown final symbol.
5) ***P***: the representative shape state transition probability matrix.
We denote it as $P = \{p_{ij}\}$, where

$$p_{ij} = P[\, s_j \text{ at } t+1 \,|\, s_i \text{ at } t \,] \quad 1 \le i, j \le N \tag{18}$$

We use the compact notation of, ***G* = (*S*, *V*, *P*)**, to indicate the complete parameter set of a stochastic grammar.

The following shows the representation of the Korean grapheme “ ㅐ ” by stochastic grammar.

$$G(\text{‘ ㅐ ’}) = (S, V, P) \tag{19}$$

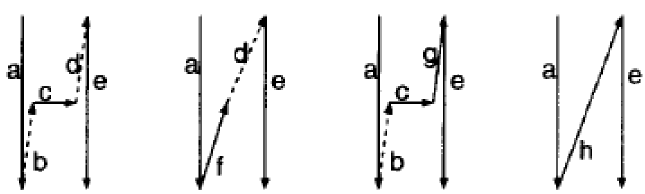


Fig. 9. Shape representation of “ ㅐ ” by primitive strokes. [7]

where

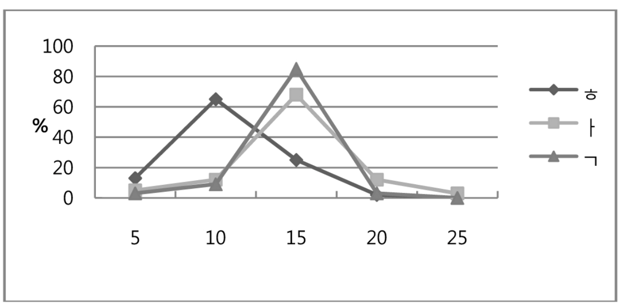
$S = \{A, B, C, D\}$ $\qquad\qquad\qquad$ $V = \{a, b, c, d, e, f, g, h, \varepsilon\}$

$$P = \begin{array}{c} \underline{A\ B\ C\ D\ \varepsilon} \\ \begin{bmatrix} 1 & 0.82 & 0.14 & 0.04 & 0 \\ 0 & 0 & 0.83 & 0.15 & 0.02 \\ 0 & 0 & 0 & 0.99 & 0.01 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

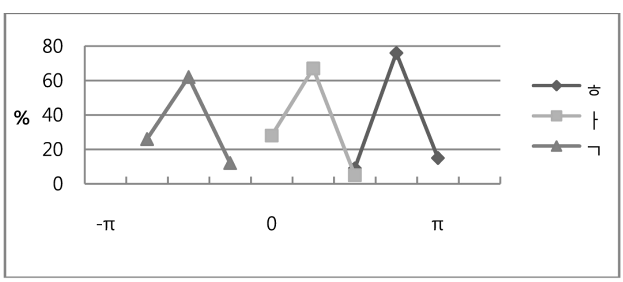
All training samples are parsed by stochastic grammar. The frequencies of corresponding state transition is counted and normalized. During verification, the accumulated log probability $P^{l}(n), 1 \leq n \leq N$ along the state sequence for each stochastic grammar l is calculated and included in ranking the candidates. If the primitive stroke is not found from the input sequence, the probability of ε is added. Thus, the missing of an important stroke will cause a large decrease of $P^{l}(n)$, which can effectively prevent the mismatching cases shown in Fig 9.

4.2.2 Position verifier

As previously stated, the global properties are excluded due to the limitation of HMM. Thus, a problem of mismatching may occur with the characters that share a high similarity of strokes and writing orders (e.g. Fig. 10). Therefore, a position verifier is proposed to deal with this problem. The main idea of a position verifier is to model the positional relationship explicitly among



(a)



(b)

Fig. 10. Percentage of (a) D and (b) θ for graphemes of character “학”

the graphemes of the character in terms of the relative position in the bounding box of the character. We first denote the center of the bounding box of the character as $p_c = (x_c, y_c)$ and the center single grapheme as $p_g = (x_g, y_g)$. Thus, the Position verifier *Pos* is defined as:

$$\boldsymbol{Pos = (D, \theta)}$$

where D is the distance between p_c and p_g, and θ is the direction from p_c to p_g.

The probabilities of D and θ for graphemes in character samples are calculated by normalized frequencies of the count of D and θ. An example of the character “학” shows the probabilities of D and θ for each grapheme, which are presented in Fig. 10.

5. Experiment and Results

The dataset used in this work is derived from the “KAIST OP2 DB,” which was created by the AI Lab at the Korean Advanced Institute of Science and Technology (KAIST). This dataset is publicly available for research use from http://ai.kaist.ac.kr/ Resource/dbase/Online/online-index.htm. This dataset includes 137,184 characters collected from more than 100 college and high school student writers. In our study, part of the dataset including 12,390 English and 45,000 Korean characters were used for training.

The first experiment was conducted to test the two sampling approaches of equidistant data points and corner detection. Table 1 shows that the model trained with the approach of equidistant data point achieved better recognition accuracy. Because more samples are obtained through equidistant data points than corner detection, the training of equidistant data points costs more than corner detection. In the following experiments, models that were only trained with equidistant data points are used for testing proposed algorithms.

In this system, the number of HMM states for the Korean character model were tuned by intuitive and empirical methods. Table 2 shows the character recognition results of these approaches. First, for every model, the fixed number of states ranging from 3 to 16 was tested, and the best performing number was chosen. Second, a half of the average length of observations in the training samples was selected for the number of the state of the class. The proportion of a half

Table 1. Recognition accuracy of the models trained with two sampling approaches

Sampling Method	Training Time (Min.)	Recognition accuracy %		
		1-candidate	2-candidate	3-candidate
Equidistant data points	53.6	87.2	91.4	94
Corner detection	47.2	79	83.6	89.2

Table 2. Recognition accuracy of the models trained with different states

State number	Fixed number	Proportion to length	Manually decided
Recognition accuracy %	87.2	90.1	88.7

was decided on empirically. Last, the number of states was chosen by intuitive knowledge. Note that the intuitive method does not show the best result due to the variety of writing styles in each class. The second method, which performed the best, was selected for comparing our proposed method.

As the clustering method was introduced for the design of multiple models, the number of generated models including both English and Korean increased more than twice as much as the single model design. In addition, the total number of states also increased by a factor of almost 2%. However, by introducing state tying, the number of states reduced by about half. As a result, the number of observation distributions became almost the same as that of the single model design. (Refer to Table 3.)

For evaluation of the proposed design method against the general design method, several tests were performed. First, to examine how well each letter and grapheme HMM was trained, the recognition tests were performed on the letter and grapheme training data. Table 4 summarizes the number of training data for each class. In Table 5 and Table 6, recognition accuracies of the letters and graphemes for two different model designs are listed. It indicates that most models were well trained, with some exceptions, and the proposed design for multiple models performed better than the single model design.

Next, the English letter recognition test was performed on the test data set of English characters (Table 5).

Finally, the recognition test was performed on the test data set of Korean characters. The result of recognition accuracy is showed in Table 8. The proposed multiple model design improved the recognition rate by almost 2% with the cost of the increased recognition time as compared with the single model design. However, the modified level building algorithm achieved an overall improvement in both recognition rate and recognition time. In addition, the shape verifier and position verifier used in post-processing also enhanced the recognition rate with a slightly extra time expense.

Table 3. Parameters that were increased by the clustering method

	Single model	Multiple HMMs	Multiple HMMs after tying
Number of models	159	324	324
Number of total states	1024	2033	1207

Table 4. Number of training data for each class

Classes	Training data number
English letter	12,390 letters
Initial consonant	45,000 graphemes
First ligature	45,000 ligatures
Middle vowel	45,000 graphemes
Second ligature	23,080 ligatures
Final consonant	23,080 graphemes

Table 5. Parameters that were increased by the clustering method

Single model design				Multiple models design			
Lowercase		Upper case		Lowercase		Upper case	
	%		%		%		%
a	88.2	A	92.0	a	92.6	A	94.1
b	87.3	B	94.8	b	88.0	B	94.8
c	85.5	C	82.5	c	87.1	C	82.8
d	90.2	D	94.0	d	90.3	D	95.8
e	94.2	E	94.4	e	96.8	E	97.3
f	96.6	F	95.0	f	97.2	F	96.6
g	85.3	G	92.1	g	90.2	G	94.5
h	90.3	H	92.1	h	92.2	H	95.1
i	70.6	I	89.3	i	89.1	I	92.3
j	86.4	J	90.2	j	92.4	J	91.4
k	84.8	K	89.9	k	85.5	K	93.1
l	89.6	L	82.6	l	89.4	L	82.0
m	92.5	M	91.0	m	92.6	M	92.7
n	90.7	N	90.3	n	91.0	N	90.5
o	83.4	O	89.7	o	83.4	O	89.8
p	88.0	P	77.8	p	92.8	P	79.1
q	89.8	Q	94.4	q	94.6	Q	94.9
r	89.1	R	92.0	r	92.3	R	92.0
s	85.0	S	87.3	s	88.0	S	88.7
t	96.3	T	94.0	t	97.2	T	94.1
u	87.5	U	90.7	u	87.7	U	90.7
v	84.6	V	88.7	v	88.8	V	88.7
w	91.6	W	91.1	w	91.0	W	91.3
x	75.8	X	84.2	x	82.9	X	87.5
y	87.5	Y	92.2	y	92.3	Y	95.0
z	76.9	Z	87.9	z	79.0	Z	87.2
Av.	87.2	Av.	90.0	Av.	90.2	Av.	91.2

6. CONCLUSION

The main goals of this research work have been well met. In this study, a multiple modeling technique based on an effective clustering approach was proposed to deal with the problem of handwriting style variations. This thesis also introduced a modified level building strategy to incorporate the Korean character combination rules. To overcome the limitations of the HMM based method, the global and structural verifiers were introduced in the post-processing procedure.

In the part of literature review for this thesis, the various recognition systems developed for both online and offline handwritten character recognition has been classified and discussed. Especially, the most popular and recent methods that are well suited for online recognition problems have been described in more detail. Special attention has also been paid to the pros and cons of the methods specified for on-line recognition of handwritten Korean characters. A brief introduction to HMM related issues has also been presented in this thesis.

Experiments showed that the proposed recognition system achieved a high writer independent recognition rate on unconstrained samples of both English and Korean characters. Compared with single HMM design, the experiments made on the multiple HMMs design revealed a significant improvement on the recognition accuracy. Even the risk of insufficient training for increased parameters caused by multiple models was nearly eliminated via state tying in the experiment. On the other hand, the experiments also showed that even though HMM based recognition methods are very well suited for the online recognition of handwritten Korean characters, some weaknesses, which are due to the inherent limitations of the HMM's assumptions, have been exposed as well. To solve this problem, the global and structural properties of the handwritten characters are included in this study. This approach has been shown to effectively improve the recognition rate.

The gap between the technical status and the required performance indicates that the problem of online handwriting recognition is not yet solved and it leaves us with future research opportunities [9]. To reach the goal of completely free handwriting recognition, we should still seriously that are being used and find ways to significantly improve the recognition performance. Based on the results of this research, a number of recommendations for future research can be made. First, an improvement can be made via the integration of multiple approaches and the joint effects of the processing step. Therefore, as has been presented in this thesis, a more reliable and efficient classifier or verifier is needed to combine with HMM based approaches to achieve a further improvement of recognition accuracy. Second, since the recognition performance depends heavily on the quality of the model database and size of sample sets, further efforts should not be spared in optimizing the model database, as well as including more valid sample sets.

REFERENCES

[1] Brown, P. (1987).The Acoustic Modeling Problem in Automatic Speech Recognition. PhD thesis, Comp. Sci. Dep., Carnegie Mellon Univ.

[2] Jaeger, S., Manke, S., Reichert, J., & Waibel, A. (2001). Online handwriting recognition: the NPen++ recognizer. International Journal on Document *Analysis and Recognition* 3 (3), 169-180.

[3] Jung, K., & Kim, H. J. (2000). On-line recognition of cursive Korean characters using graph representation. Pattern Recognition 33, 399-412.

[4] Kim, H. Y., & Kim, J. H. (2001). Hierarchical random graph representation of handwritten character and its application to Hangul recognition. *Pattern Recognition* 34 (2), 187-201.

[5] Kim, P. K., & Kim, H. J. (1995). On-line recognition of run-on Korean characters. *Proc. 3rd IEEE Int. Conf. on Document Analysis and Recognition '95*, Vol.1, Montreal, Canada, pp.54-57.

[6] Kim, S. K., Kim, J. W., & Kim, H. J. (1996) .On-line recognition of cursive Korean characters using neural networks. *Neurocomputing* 10(3): 291-305

[7] Kwon, J. O., Sin, B. K., & Kim, J. H. (1997). Recognition of On-line Cursive Korean Characters Combining Statistical and Structural Models. *Pattern Recognition*, 30, 8, pp.1255-1263

[8] Kwon, O. S., & Kwon, Y. B. (1993). An on-line recognition of Hangul handwriting using dynamic generation of line segments. *Proceeding of The Twentieth KISS Spring Conference*, pp.151-154.

[9] Liu C. L., Jaeger, S., & Nakagawa, M. (2004). Online recognition of Chinese characters: the state-of-the-art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.26, No.2.

[10] Park, H. S., & Lee, S. W. (1996). Off-line recognition of large-set handwritten characters with multiple hidden Markov models, *Pattern Recognition* 29(2), 231-244.

[11] Parisse, C. (1996). Global word shape processing in on-line recognition of handwriting. *IEEE Trans. Pattern Anal. Mach. Intell*, 18 (4) 460-464.

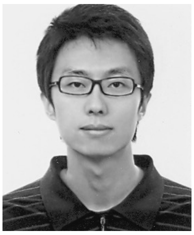
[12] Plamondon, R., & Srihari, S. N. (2000). On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* , 22 (1), 63-84.

[13] Rabiner, L. R., & Levinson, S. E. (1985). A speaker-independent, syntax-directed, connected word recognition system based on hidden Markov models and level building. *IEEE Trans. Acousf. Speech Signal Process*., 33, (3), pp.561-573.

[14] Senior, A. W., & Robinson, A. J. (1998). An on-line cursive handwriting recognition system, *IEEE Trans. Pattern Anal. Mach. Intell*, 20 (3) 309-321.

[15] Vuori, V. (2002). Adaptive Methods for On-Line Recognition of Isolated Handwritten Characters, PHD dissertation

[16] Yacoubi, A. E., Gilloux, M., Sabourin, R., & Suen, C. (1999). An HMM-based approach for on-line unconstrained handwritten word modeling and recognition. *IEEE Trans. Pattern Anal. Mach. Intel,* 21 (8) 752-760.



Ming Ma

He received the Ph.D Degree in Info. & Comm. Eng. from Paichai University in 2009. He received the M.S Degree in Computer Science from PAICHAI University in 2004. His research interests include Information Recognition and Multimedia.

Dong-Won Park

He received the Ph.D Degree in Computer Science from Texas A&M University in 1993. He has been a professor at Paichai University since 1994. His research interests include Networked Multimedia and Embedded S/W.

Soo Kyun Kim

He received Ph.D. in Department of Computer Science & Engineering from Korea University in 2006. He joined Telecommunication R&D center at Samsung Electronics Co., Ltd., from 2006 and 2008. He is now a professor at Department of Game Engineering at Paichai University, Korea. His research interests include multimedia, pattern recognition, image processing, mobile graphics, geometric modeling, and interactive computer graphics.

Syungog An

She received the Ph.D Degree in Computer Science from Korea University in 1989. She has been a professor at Paichai University since 1991. Her research interests include Computer Graphics, DataBase and Game Development.