# Data-driven Design of HMM Topology for On-line Handwriting Recognition

JAY J. LEE, JAHWAN KIM, and JIN H. KIM

*Dept. of Electrical Engineering & Computer Science, KAIST*

*373-1, Kusong-dong, Yusong-gu, Taejon, 305-701, Korea*

*E-mail: { joony, jahwan, jkim }@ai.kaist.ac.kr*

## Abstract

Although HMM is widely used for on-line handwriting recognition, there is no simple and well-established method of designing the HMM topology. We propose a data-driven systematic method to design HMM topology. Data samples in a single pattern class are structurally simplified into a sequence of straight-line segments, and then these simplified representations of the samples are clustered. An HMM is constructed for each of these clusters, by assigning a state to each straight-line segments. Then the resulting multiple models of the class are combined to form an architecture of a *multiple parallel-path HMM*, which behaves as a single HMM. To avoid excessive growing of the number of the states, parameter tying is applied in that structural similarity among patterns is reflected. Experiments on on-line Hangul recognition showed about 19 % of error reductions, compared to the intuitive design method.

*Keywords*: On-line handwriting recognition, hidden Markov model, data-driven topology design, multiple parallel-path HMM, state-tying based on structural similarity, Hangul recognition

## 1. INTRODUCTION

As one of the major research directions for on-line handwriting recognition, hidden Markov model (HMM) is widely used because of the time sequential nature of on-line scripts as well as its capability of modeling shape variability in probabilistic terms. However, there has been no serious study or guidance in the design of HMM topology. Previous studies suggested that HMM should be designed depending on the signal being modeled.[1,2] The model needs to have enough number of free parameters to accommodate complexity of target patterns and to reflect properties of the patterns. In practice, however, an arbitrary increment of the model parameters is not recommended, since available training samples are usually limited. Therefore, HMM topology should be determined based both on available training data and on the target patterns to be represented.

In this paper, we are focusing on two design parameters, *i.e.*, the number of states in HMM and the number of models for a class. Despite its importance, relatively little attention has been paid to the design of HMM topology. Simply, the same number of states was used,[3] or all possible instances of the number of states were tested and then the best one was selected.[4] Attempts have been also made to incorporate intuitive knowledge.[5, 6] Data-driven methods include utilization of length information such as average length of observation symbols [7] or length of pre-defined sub-character units,[8] and more systematic adjustment of the number of states by successive state-splitting [9] or merging [10] according to maximum likelihood criterion. Structural design method was also presented,[11] where the topology of a Markov network was inferred from a finite set of samples.

We propose a data-driven method of design HMM topology for on-line handwriting recognition. Our design principle is that HMM topology should be constructed from data, reflecting the structure of a target pattern. Here, we assumed that a target pattern is composed of straight-line segments. Accordingly, a sample of the target pattern can be structurally decomposed and simplified as a sequence of straight-line segments, i.e., *structural units*. According to our design principle, the HMM has a state corresponding to each straight-line segment. To handle shape and writing-order variations present inside a class, sequences of straight-line segments, which are simplified representations of samples, are clustered to construct multiple models. The resulting multiple models for a single class are combined to form a single HMM architecture, called a *multiple parallel-path HMM*. For training, the initial observation probability distribution for each state is estimated from the distribution of corresponding straight-line segments, and then the Viterbi path training method is applied. When models for a single class have parts that are not simply similar in shape but *structurally* similar, corresponding states are tied. The number of parameters in the HMM are hence reduced to a manageable size. This state-tying based on structural similarity is performed at the design stage of the HMM.

The proposed method was evaluated on on-line Hangul (the Korean script) recognition, since Hangul graphemes are typically structured with line segments. Experiments showed that our method reduced about 19 % of character recognition error compared to the intuitive design methods. We believe that the proposed design method can be applied to other scripts that are mostly composed of straight-line segments, such as Chinese characters.

The organization of the paper is as follows. Section 2 presents the data-driven design method of HMM topology, combining architecture of multiple models, and the structural state-tying method. Section 3 introduces Hangul and the on-line Hangul recognition system briefly, then addresses the external duration modeling for performance improvement. Section 4 shows the experimental results of the proposed approach and analysis of the results. Conclusion is followed in Sect. 5.

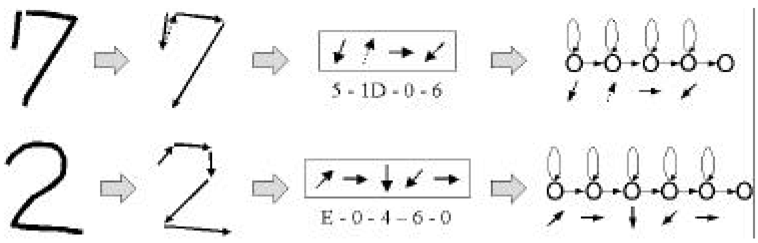
## 2. DATA-DRIVEN DESIGN OF HMM TOPOLOGY

In this section, we will describe how to determine the number of states in HMM and the number of models for each pattern class, based on training samples. We will also explain how these multiple models for a single class are combined in the architecture of a *multiple parallel-path HMM*. Finally, the structure-based state-tying method to reduce the number of parameters will be explained.

## 2.1. Mapping line segment to HMM state

The number of HMM states is an important design parameter. It is a measure, albeit crude, of the complexity of the finite state grammar represented by the Markov chain embedded in the HMM [6] and corresponds, roughly, to the number and dynamics of signal ‘prototypes’ being modeled. For instance, a state could correspond to certain phonetic event in a speech recognition system.[2] Thus, in modeling complex patterns, the number of states should be increased accordingly. When there are insufficient numbers of states, the discrimination power of the HMM is reduced, since more than one signal should be modeled on one state. On the other hand, the excessive number of states can generate the over-fitting problem when the number of training samples is insufficient compared to that of the model parameters.[12]

We propose a data-driven design method of HMM topology for on-line handwriting recognition. In this method, the number of HMM states is determined by the structural decomposition of the target pattern. Handwriting is structurally simplified as a sequence of the straight-line segments (see Fig. 1-(b)). After noise removal and smoothing operation, adjacent pen movements with similar directions are grouped into a single straight-line segment. An invisible pen-up movement between pen-down strokes is also inserted as an imaginary line. Average direction of a line segment with pen-down movement is encoded as one of 16-direction codes, and the imaginary line is encoded by another 16-direction codes as shown in Fig. 2. We call the resulting direction code sequence a *skeleton pattern* (Fig. 1-(c)). It is regarded as a simplified representation of the pen movement, since both hand vibrations and length variations are ignored but only directional information remains. It describes a time-sequential and global shape of the pen movements.

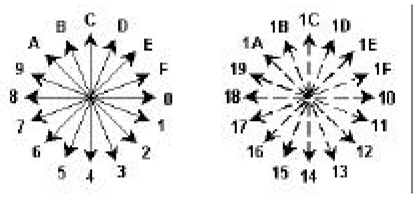



(a) (b) (c) (d)

Fig. 1. Examples of HMM topology design

(a) Handwriting sample, (b) Line segment approximation, (c) Skeleton pattern, (d) Resulting HMM

The structure of HMM is based on the skeleton pattern of a sample as shown in Fig. 1-(d). The transition structure of the model is a simple left-to-right casual model. The number of states is determined by mapping each straight-line segment into a single HMM state. Thus, each state assumes a uni-modal feature distribution only for the corresponding straight-line. Length variations of the straight-line are modeled in the self-loop of the state. As a consequence, each state of HMM corresponds to a straight-line segment of handwriting in time-sequential order. For this reason, external knowledge can be utilized for the verification of the recognition result.




(a) Pen-down codes (b) Pen-up codes

Fig. 2. Two sets of 16 direction codes for a skeleton pattern

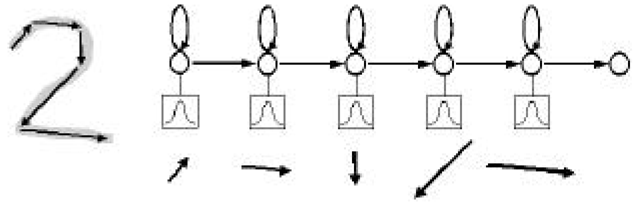


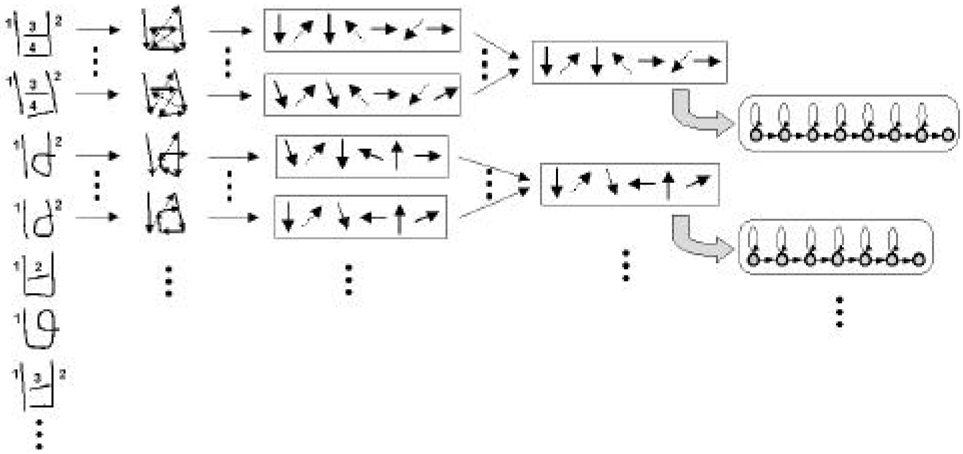
Fig. 3. Initial observation probability distribution from a skeleton pattern

Since the maximum likelihood training method of HMM is a kind of the steepest gradient search method, a good initial estimate, instead of random or uniform probability, is helpful for finding the global maximum of the likelihood function.[1] We can obtain initial observation probability distributions from the mapping relation between states and the skeleton pattern (see Fig. 3). The distribution of the line segments is accumulated from the training sample, and then its normalization is used as the initial parameter of the corresponding state. When good initial observation probability distributions are given, the Viterbi path training methods[13] works better and faster, compared to the usual Baum-Welch method.

## 2.2. Design of multiple models by clustering

Whether or not having a single HMM to model whole patterns of a single class is also an important design decision. Several models may be needed to represent quite different shapes of a single class, such as the well-known division of printed style and cursive style (see Fig. 4). To this end, a clustering method is usually applied to obtain multiple models in each class. Nonetheless, it is difficult to decide the number of models in advance for top-down clustering. Similarly, selecting a proper distance measure for clustering samples without any additional knowledge or constraints is a difficult task.

Fig. 4. Two different writing styles of the Roman character ‘b’



(a) (b) (c) (d) (e)

Fig. 5. Example of a multiple HMM design: Hangul consonant 'ㅂ'

(a) Various handwriting samples, (b) Line segment approximations, (c) Skeleton patterns, (d) Representative patterns, (e) Resulting HMMs

Skeleton patterns within a class, each of which is the simplified representations of sample, are clustered to determine the number of models for the class. The agglo merative clustering method,[12] which is a bottom-up approach, gathers skeleton patterns of the similar direction codes into a cluster. Since the skeleton patterns contain only principal pen movements, the proposed method collects the data samples of similar global shape as shown in Fig. 5-(c) and (d).

For the clustering, the distance between two skeleton patterns are defined as follows. The distance $D(X^1, X^2)$ between two skeleton patterns $X^1 = (x_1^1, x_2^1, \ldots, x_{L_1}^1)$ and $X^2 = (x_1^2, x_2^2, \ldots, x_{L_2}^2)$ is computed by the *dynamic programming* using the recursion relation:

$$D(x_i^1, x_j^2) = d(x_i^1, x_j^2) + \min\{ D(x_{i-1}^1, x_j^2), D(x_{i-1}^1, x_{j-1}^2), D(x_i^1, x_{j-1}^2)\},$$

$$D(X^1, X^2) = D(x_{L_1}^1, x_{L_2}^2)$$

where $d(x_i^1, x_j^2)$ is the directional difference of direction code $x_i^1$ and $x_j^2$.

The skeleton pattern that appears most frequently in each cluster is chosen as the *representative pattern* of that cluster (Fig. 5-(d)). Thus, various writing styles of a class are reflected by the set of representative patterns. An HMM is, then, constructed from each representative pattern by the method described in Sect. 2.1. As a consequence, the number of representative patterns decides the number of models in a class, and the length of the representative pattern determines the number of states of the corresponding HMM. In our experiment, clusters containing only a small number of samples are disregarded to prevent generating too many models.

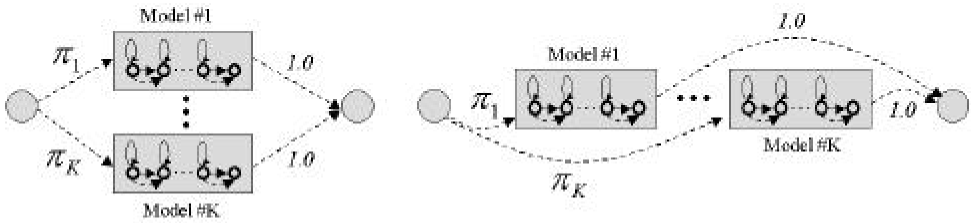
## 2.3. Combining models to one multiple parallel-path HMM

One model for one class yields many benefits. It allows a modular design of a recognizer in that a model can be replaced with another easily. If different numbers of models exist in each class, models of the same class may compete for selection. It may hurt our attempt to select the top most labels for post-processing. In addition, *a prior* probability, which is obtained from language corpus, cannot be easily applied if there exist multiple models for a single class label.

For these reasons, we propose a *multiple parallel-path HMM (MPP-HMM)* architecture, to combine the multiple models of the same class into a single HMM structure. Dummy initial and dummy final nodes are introduced and connected to the multiple models of the same class. Then, these models are arranged in parallel. There is no connection between the multiple models. Thus, each constituent model forms one of the multiple paths from the dummy initial node to the dummy final one (see Fig. 6-(a)). *A prior* probability of each constituent model, Pr$(\lambda_i)$, is assigned to the initial probability of the model $\pi_i$:

$$\pi_i = \Pr(\lambda_i) = K_i / K, \qquad \sum_{i=1}^{C} \pi_i = 1.0,$$

where $\lambda_i$ denotes the constituent HMM for cluster $i$, $K_i$ is the size of data in cluster $i$, $K$ is the size of all samples for the class, and $C$ is the number of clusters in the class. Even though the MPP-HMM contains the multiple models inside, it behaves like a single HMM for the class. The structure of Fig. 6-(a) can be represented to the structure of Fig. 6-(b), which equals to the general left-to-right HMM structure.




(a) (b)

Fig. 6. Architecture of multiple parallel paths HMM

The forward-backward algorithm of HMM still holds in this architecture. The model likelihood score $P(O|\lambda)$ for input $O$ and HMM $\lambda$ is calculated by summing all of constituent models' likelihood, each of which is weighted by corresponding *a prior* probability $\pi_i$:

$$P(O \mid \lambda) = \sum_{i=1}^{C} P(O \mid \lambda_i) \times \pi_i$$

The Viterbi search for finding the maximum likelihood path also can be applied without modification. It is the selection of the maximum path $Q^*$ among those for constituent models:

$$Q_i^* = \arg\max_{Q_i} P(O \mid Q_i, \lambda_i) \times P(Q_i \mid \lambda_i),$$

$$i^\wedge = \arg\max_i (P(O \mid Q_i^*, \lambda_i) \times P(Q_i^* \mid \lambda_i) \times \pi_i),$$

$$Q^* = Q_{i^\wedge}^*$$

where $Q_i$ is a state sequence of model $\lambda_i$.

Training for the MPP-HMM involved two sessions. In the first session, each constituent model is trained individually with only the data inside the corresponding cluster. Note that, samples from the small size clusters, which are disregarded during the model building process, are disregarded in this session. Instead, they are included in the next training session. Each of them selects the maximum likelihood model among the multiple models, and then participates in the update of the parameters of the selected model. Finally, new *a priori* probability of each model is calculated according to the number of updated training samples, and then used for the path probability to the model.

## 2.4. State-tying based on structural similarity

Maintaining a balance of design between the model complexity and the amount of training data is critical for successful recognition system. Even though multiple models dramatically increase the modeling power, the limited number of training samples is not enough for deciding the large number of free parameters. Parameter-tying is one of the solutions for maintaining multiple models with limited training samples. Parameter-tying with HMM-based modeling is usually applied to states, actually observation probability distributions.[2,14]

Due to fast and sloppy writing, the pen movement is easily affected by the previous and the following pen directions, and therefore, simple shape similarity may not be robust for different writers. Thus, in our method, not only the local closeness of output distributions but also their *structural* similarities are considered to determine the state-tying. The structural similarity is measured from the relative position of observation inside a pattern and from the global shape of the pattern. We can easily measure the structural similarity by comparing representative patterns, which are used for building multiple HMMs in Sect. 2.2. In particular, our method determines which states are tied at the design phase.

The structural state-tying method is applied only to the states in the same class. The *edit distance* [15] is applied for comparing representative patterns. The edit distance measures the distance between two strings by the minimum cost sequence of ' edit operations' needed to change the one string into the other. The edit distance $D_E$ $(X, Y)$ between two representative patterns $X = (x_1, x_2, \ldots, x_{L_1})$ and $Y = (y_1, y_2, \ldots, y_{L_2})$ is computed by the *dynamic programming* method using the recursion relation:

$$D_E(x_i, y_j) = \min\{\ D_E(x_{i-1}, y_{j-1}) + \boldsymbol{d}(x_i, y_j) \times E_S,$$

$$D_E(x_{i-1}, y_j) + E_D,\quad D_E(x_i, y_{j-1}) + E_I\}$$

where $E_s$ is a substitution penalty, $E_D$ is a deletion penalty, $E_I$ is an insertion penalty, and

$$\boldsymbol{d}(x_i, y_j) = 0, \quad \text{if} \quad d\,(x_i, y_j) < T$$

$$\boldsymbol{d}(x_i, y_j) = 1, \quad \text{otherwise}$$

where $d\ (x_i, y_j)$ is the directional difference between direction code $x_i$ and $y_j$, and $T$ is the predefined threshold. Then,

$$D_E(X, Y) = D_E(x_{L_1}, y_{L_2})$$

If the distance $D_E\ (X,\ Y)$ is within a threshold, $X$ and $Y$ are considered for tying. Among the aligned matching result of these two patterns, state $i$ of the model for $X$ and state $j$ of the model for $Y$ are tied when $\boldsymbol{d}(x_i, y_j) = 0$. Consequently, observation probability distributions are shared among the tied states, but transition probabilities are separately maintained.

Figure 7 shows an example of the structural state-tying. The vertical line corresponding to the 1st state of a left model and the vertical line corresponding to the 1st state of a right model are tied. However, the same directional vertical line corresponding to the 3rd state of the right model is not tied to the 1st state of the left model because their relative positions are different.

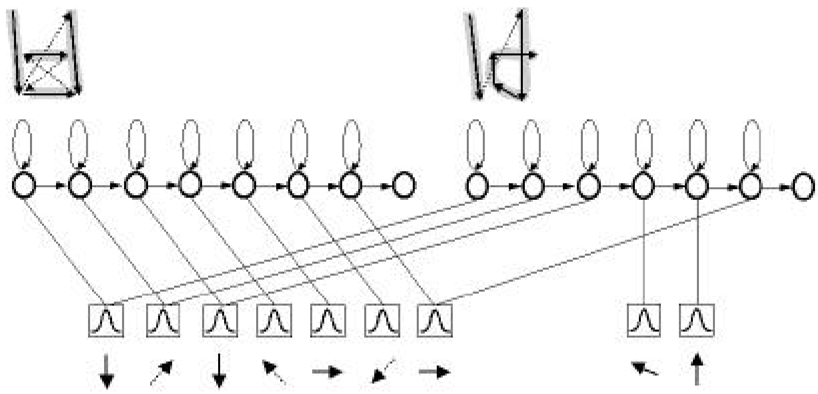


Fig. 7. Example of state-tying based on structural similarity

## 3. IMPLEMENTATION FOR ON-LINE HANGUL RECOGNITION

The proposed design method was evaluated on on-line Hangul recognition system. In this section, Hangul and on-line Hangul recognition system will be explained briefly. Also presented is the heuristic of external duration modeling which was introduced to achieve a level of duration modeling with small computational burden.

## 3.1. Hangul

*Hangul*, the script used for Korean language, is a phonetic writing system. A character in Hangul, which corresponds to a single spoken syllable, is formed by spatial arrangement of either two or three graphemes: an initial consonant, a vowel, and an optional final consonant. Graphemes consist of 24 basic phonetic symbols. Characters are usually written in the order of an initial consonant, a vowel, and a final consonant, if any. Each grapheme is formed by line segments consisting of sequential combination of horizontal, vertical, and/or diagonal lines (see Fig. 8). Especially in the case of vowel, a long vertical line with short horizontal lines or a long horizontal line with short vertical lines constitutes basic vowels. Additional consonants/vowels are from spatial combinations of the basic consonants/vowels. Only two consonants with circles, 'ㅇ' and 'ㅎ', are exceptions. Therefore, a Hangul character can be easily decomposed and represented as a sequence of straight-line segments, although co-articulating effects deform the basic shapes. Several previous structural Hangul recognizers attempt to extract these basic line segments to recognize characters.[16,17]

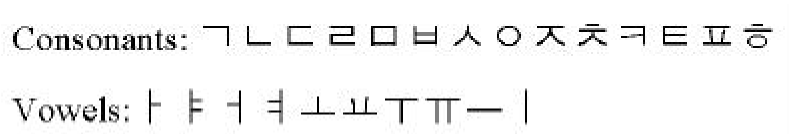



Fig. 8. 24 basic graphemes of Hangul

## 3.2. HMM based on-line Hangul recognizer

We have reported a HMM network-based approach for on-line Hangul recognition system.[5] Discrete HMM was adopted to construct the grapheme and the ligature models. Invisible pen-up movements as well as conventional pen-down strokes were encoded into observation symbols that consist of two sets of 16 direction codes. A character in Hangul can be considered as an alternating sequence of graphemes and intermediate ligatures. By sequential concatenation of these two kinds of HMM as the order of writing character, a 5-layer finite state network called *BongNet* was designed for all legal characters as shown in Fig. 9. The first, the third, and the fifth layers model the initial consonant, the vowel, and the final consonant, respectively. The second and the fourth layers correspond to ligatures between graphemes. All possible ligatures were grouped using the information of character's topology or the spatial arrangement of component graphemes.

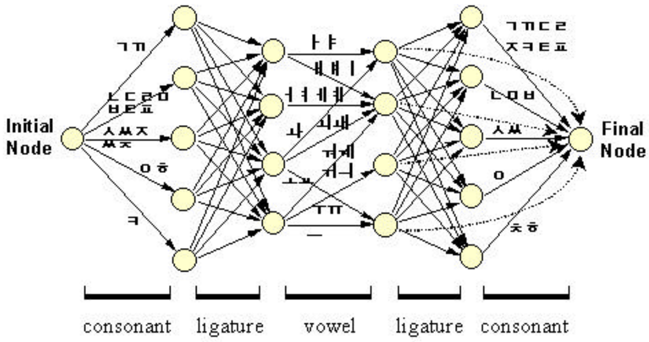
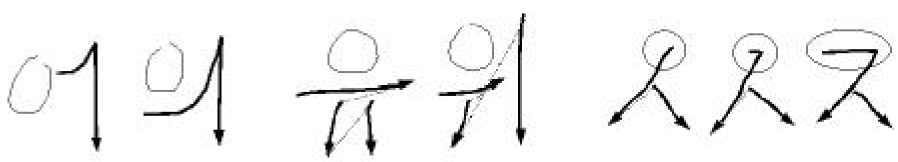



Fig. 9. *BongNet* – Hangul character recognition network

In such a finite state network, a path spanning from the initial node to the final node, represents a character. The recognition problem is a matter of finding the most likely path that best aligns to an input code sequence. We used a modified version of the Viterbi algorithm, which keeps several highest hypotheses during search. Checking language models or other source of knowledge on post-processing reorders candidates. Besides language models, structural knowledge from shape analyzers, position analyzers, and pair-wise discriminators were applied.[18] From the maximal probability path, optimal character and ligature segmentation, associated character labels are obtained simultaneously. Any external segmentation is not necessary because the boundaries are obtained from a global viewpoint in recognition.

## 3.3. External state duration modeling

States of normal HMM have exponential duration density inherently.[1] Such exponential duration characteristic is inappropriate for most physical signals. Particularly for our on-line recognition problem, the durational information of the straight-line segments was not reflected on the proposed topology design method. Patterns with similar pen movements but largely different in their lengths can be often confused. Hangul has several confusing classes that have almost same pen movements as those of other classes (see Fig. 10-(a), (b)). Sometimes, handwritten input with a small noisy hook or serif may be mistaken to other class when no duration information is used (see Fig. 10-(c)).



(a) ‘ㅓ’ Vs. ‘ㅢ’ (b) ‘ㅠ’ Vs. ‘ㅟ’ (c) ‘ㅅ’ Vs. ‘ㅅ’ Vs. ‘ㅈ’

Fig. 10. Examples of confusing Hangul graphemes by similar pen movements

We have applied the external state duration method[19] in order to emphasize duration difference. This method adds random variables to the state duration outside the model. After model training, the duration information is calculated from the maximal paths of all training samples. For recognition purposes, it is used as a post-processor. After the Viterbi search algorithm determines the maximum likelihood state sequences, the duration $d_i$ of state $i$ is counted along the candidate Viterbi paths, and then the duration score is added as the post-processing score:

$$\hat{\boldsymbol{d}} = \boldsymbol{d} \times \left(\prod_{i=1}^{N} P_i(d_i)\right)^a$$

where $\boldsymbol{d}$ is the model likelihood score on the maximum path, $P_i(d_i)$ is the duration score of duration $d_i$ on state $i$, $N$ is the number of states, and $\boldsymbol{a}$ is the multiplier factor to adjust the importance of the duration part of the scoring. Since the state duration is not estimated as part of the training session, it is not the maximum likelihood estimate in the strict sense, but a heuristic one. Even though it does not guarantee an optimal solution, it generally works well. This explicit duration model needs a negligible amount of the computational overhead compared to the usual parametric or non-parametric internal state duration model. However, it may be unprofitable in case of a single model for a single class, because one state may model different pen movements, and thus, hold various state durations.

## 4. EXPERIMENTS

The data for experiments were collected without any constraint on writing style. Both printed and cursive handwriting styles were included. To train HMMs, we used about 85,000 Hangul characters in frequently used 2,350 Hangul character classes written by 84 writers. Since grapheme boundaries in these characters are not available, graphemes were manually segmented for the training. The test data were collected from different writers who did not participate in the training data. Two kinds of texts were used that are composed of 580 and 168 kinds of characters respectively. Test data consists of 12,140 Hangul characters written by 23 writers.

For comparing the proposed design method against an ordinary design method, the Hangul recognition system with a single HMM for each class [5] was used. In this system, the number of HMM states were tuned by intuitive and empirical methods. Table 1 shows the character recognition results of these approaches. First, the fixed number of states for every model, ranging from 3 to 16, was tested, and then 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, a half, was decided 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 writing styles in each class. The best performing, the second method is selected for comparison with our proposed method.

Table 1. Correct recognition rate (%) of the character recognition test : single model per class

| Methods | Fixed number of states | Proportion of code length | Manual decision |
|---|---|---|---|
| Correct recognition rate | 90.77 % | 91.55 % | 91.54 % |

Next, HMM topology was decided by the proposed method. Table 2 shows that the number of models generated was increased about 2.5 times with our method, compared to the single model HMM (an average of 2.5 models per class). The total number of states was also increased by a factor of about 2. However, the state-tying method reduced them to about a half. As a result, the number of free parameters for the observation distributions becomes almost the same as that of the single model setting.

Table 2. Increased parameters by the proposed method

| | Single model | MPP-HMM | MPP-HMM after tying |
|---|---|---|---|
| Number of models | 107 | 258 | 258 |
| Number of different states | 577 | 1070 | 598 |

MPP-HMM : Multiple Parallel-Path HMM

Two kinds of tests were performed for evaluation of the performance of the proposed method. First, to examine how well each grapheme HMM was trained, recognition tests were performed for grapheme training data: 83,151 isolated initial consonants of 19 classes, 83,536 vowels of 21 classes, and 49,431 final consonants of 27 classes. The second column of Table 3, labeled 'Grapheme test', shows the recognition rates of these tests. Note that the use of the state duration model gives remarkable performance improvement, *i.e.*, about 40 % error reduction. This improvement is due to the durational difference among confusing graphemes. Next, Hangul character recognition test was performed by the test data set of Hangul characters. The third column of Table 3, labeled 'Character test', shows the recognition results. About 19 % of recognition errors were reduced by the proposed method compared to the single model setting. Note that state-tying did not degrade the recognition accuracy.

Table 3. Correct recognition rate (%) of grapheme and character recognition

| | Grapheme test | Character test |
|---|---|---|
| Single Model | 93.85 % | 91.55 % |
| MPP-HMM | 93.21 % | 92.04 % |
| MPP-HMM + D | 96.56 % | 93.11 % |
| MPP-HMM + D + T | 96.30 % | 93.16 % |

D : external duration model, T : structural state-tying

Table 4 shows the increment of time complexity due to multiple models. Since search space of the recognition network broadened by the multiple model setting, average time for character recognition was increased about 2 times.

Table 4. Increased time complexity by multiple model setting (in SUN SPARC-II workstation)

| | Single model | MPP-HMM |
|---|---|---|
| Avg. recognition time | 0.10 (sec/char) | 0.19 (sec/char) |

## 5. CONCLUSION

A data-driven systematic design method of HMM topology for on-line handwriting recognition was proposed. The number of models in each class and the number of HMM states in each model were determined by the structurally simplified sequences of the straight-line segments and their clusters. As a result, different handwriting styles were modeled by multiple HMMs, and their states were forced to correspond to the line segments of the target pattern in time sequential order. The multiple models in a class were combined in parallel to form the structure of the multiple parallel-path HMM, and then it behaves as a single HMM. States with structural similarity were tied, hence the number of HMM parameters were reduced. For the practical application of the system, the external states duration modeling was applied. The experiments to on-line Hangul handwriting recognition showed that the proposed method reduced about 19% of the error rate compared to the intuitive design methods.

## ACKNOWLEDGEMENT

The authors would like to thank Prof. Yeungwoo Choi from Sookmyung Women's Univ., Korea and Prof. Bong-Kee Sin from Pukyung Nat. Univ., Korea for discussions regarding this work.

## REFERENCES

1. L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, 1993
2. Kai-Fu Lee, *Automatic Speech Recognition*, Kluwer Academic Publishers, 1989
3. Jin-Young Ha, Se-Chang Oh, and Jin H. Kim, "Recognition of Unconstrained Handwritten English Words with Character and Ligature Modeling", *Int. Journal of Pattern Recognition and Artificial Intelligence*, **9**, 3 (1995) 535-556
4. G. Rigoll, A. Kosmala, J. Rottland, and Ch. Neukirchen, "A Comparison Between Continuous and Discrete Density Hidden Markov Models for Cursive Handwriting Recognition", *Proc. of Int. Conf. on Pattern Recognition*, Vienna, 1996, 205-209
5. Bong-Kee Sin and Jin H. Kim, "Ligature Modeling for Online Cursive Script Recognition", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **19**, 6 (1997) 623-633
6. P. F. Brown, *The Acoustic-Modeling Problem in Automatic Speech Recognition*, Ph. D. thesis, Dept. of Computer Science, Carnegie Mellon University, 1987

7. Krishna S. Nathan, Homayoon S. M. Beigi, Jayashree Subrahmonia, Gregory J. Clary, and Hiroshi Maruyama, "Real-Time On-Line Unconstrained Handwriting Recognition Using Statistical Methods", *Proc. of Int. Conf. on Acoustics, Speech, and Signal Processing*, Detroit, Michigan, 1995, 2619-2622
8. Jianying Hu and Michael K. Brown, "HMM Based On-Line Handwriting Recognition", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **18**, 10 (1996) 1039-1045
9. Jun-ichi Takami and Shigeki Sagayama, "A Successive State Splitting Algorithm for Efficient Allophone Modeling", *Proc. of Int. Conf. on Acoustics, Speech, and Signal Processing*, San Francisco, 1992, I.573-576
10. Andreas Stolcke and Stephen M. Omohundro, *Best-first Model Merging for Hidden Markov Model Induction*, Technical Report, TR-94-003, International Computer Science Institute, Berkeley, 1994
11. Michael G. Thomason and Erik Granum, "Dynamic Programming Inference of Markov Networks from Finite Sets of Sample Strings", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **8**, 4 (1986) 491-501
12. Richard O. Duda and Peter E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, Inc., 1973
13. L. R. Rabiner, J. G. Wilpon, and B. H. Juang, "A Segmental k-means Training Procedure for Connected Word Recognition", *AT&T Technical Journal*, **65**, 3 (1986) 21-31
14. S. J. Young and P. C. Woodland, "The Use of State Tying In Continuous Speech Recognition", *Proc. of EUROSPEECH '93, the 3rd European Conf. on Speech Communication and Technology*, Berlin, Germany, 1993, 2203-2206
15. Robert A. Wagner and Michael J. Fischer, "The String-to-String Correction Problem", *Journal of the Association for Computing Machinery*, **21**, 1 (1974) 168-173
16. Ho-Yon Kim and Jin H. Kim, "Hierarchical Random Graph Representation of Handwritten Characters and its Application to Hangul Recognition", *Pattern Recognition*, **34**, 2 (2001) 187-201
17. Hang Joon Kim and Pyeoung Kee Kim, "Recognition of Off-line Handwritten Korean Characters", *Pattern Recognition*, **29**, 2 (1996) 245-254
18. Jae-Ook Kwon, Bong-Kee Sin, and Jin H. Kim, "Recognition of On-line Cursive Korean Characters Combining Statistical and Structural Methods", *Pattern Recognition*, **30**, 8 (1997) 1255-1263
19. L. R. Rabiner, B. H. Juang, S. E. Levinson, and M. M. Sondhi, "Recognition of Isolated Digits Using Hidden Markov Models with Continuous Mixture Densities", *AT&T Technical Journal*, **64**, 6 (1985) 1211-1234