

Substroke Approach to HMM-based On-line Kanji Handwriting Recognition

Mitsuru NAKAI, Naoto AKIRA, Hiroshi SHIMODAIRA and Shigeki SAGAYAMA*

Graduate School of Information Science,
Japan Advanced Institute of Science and Technology
{mit,n-akira,sim,sagayama}@jaist.ac.jp

*Graduate School of Information Science and Technology,
The University of Tokyo
sagayama@hi1.t.u-tokyo.ac.jp

Abstract

A new method is proposed for on-line handwriting recognition of Kanji characters. The method employs substroke HMMs as minimum units to constitute Japanese Kanji characters and utilizes the direction of pen motion. The main motivation is to fully utilize the continuous speech recognition algorithm by relating sentence speech to Kanji character, phonemes to substrokes, and grammar to Kanji structure. The proposed system consists input feature analysis, substroke HMMs, a character structure dictionary and a decoder. The present approach has the following advantages over the conventional methods that employ whole character HMMs. 1) Much smaller memory requirement for dictionary and models. 2) Fast recognition by employing efficient substroke network search. 3) Capability of recognizing characters not included in the training data if defined as a sequence of substrokes in the dictionary. 4) Capability of recognizing characters written by various different stroke orders with multiple definitions per one character in the dictionary. 5) Easiness in HMM adaptation to the user with a few sample character data.

1. Introduction

On-line handwriting recognition is a problem to decode a time sequential pattern of pen movement that can be considered as the same problem as speech recognition. HMM (Hidden Markov Model) is the most popular technique for speech recognition and, in recent years, has been successfully applied to European and American on-line character recognition[1, 4, 6, 8]. In this case, each whole letter of the alphabet is modeled typically by one HMM, and all words are represented by employing only some dozens of HMMs at most. On the other hand, applying HMMs to Japanese characters is not a simple problem as they include more than 6,000 Kanji (Chinese-original) characters as well as Hiragana, Katakana, numerics and other character sets. In past studies, HMM was introduced into Hiragana recognition with 46 classes[2, 7], and educational Kanji character

recognition with 881 classes[3, 9], in which each character was independently modeled by an HMM. In this simple approach, the total size of models was proportional to the number of characters to recognize. Even covering a small subset of Japanese characters, one study[9] required the model size of approximately 2MB and considered to be already too large for small-sized applications, such as PDAs. Moreover, the required model size would increase if we take into account a number of possible variations of stroke orders of writing Kanji, since they are composed of up to 30 strokes for each and may be written in writer's own stroke orders.

In this paper, we propose the use of substroke HMMs for on-line handwriting recognition that can handle a large number of Kanji characters with a small number of substroke HMMs modeling common segments in characters in combination with a dictionary defining the character structures. This new approach was highly motivated by continuous speech recognition: on-line character recognition was considered as recognition of a sequence of strokes or segments in which the search space is defined by Kanji substroke dictionary. We define a hierarchical dictionary based on "substrokes (direction line segments)", "strokes (connected pen-down segments)", "sub-Kanji characters (radicals)" and "Kanji character". In comparison of this structure with continuous speech recognition, "stroke" corresponds to "phoneme", "character" to "word speech" and "written sentence" to "sentence speech". In addition, we employ continuous output HMMs with Gaussian distributions instead of quantized feature of 8 or 16 directions chain codes used in most conventional methods. The advantages of substroke-based HMM are summarized as follows.

1. The memory required for models and dictionary is small. In case of 6,353 Kanji characters (total of JIS X0208-1983 Kanji level 1 and level 2), the memory requirement is approximately as 1/40 smaller as whole character HMMs.
2. Recognition speed is improved by using efficient substroke network search. Even in case of full search without beam pruning, it is nearly five times faster than character HMM.

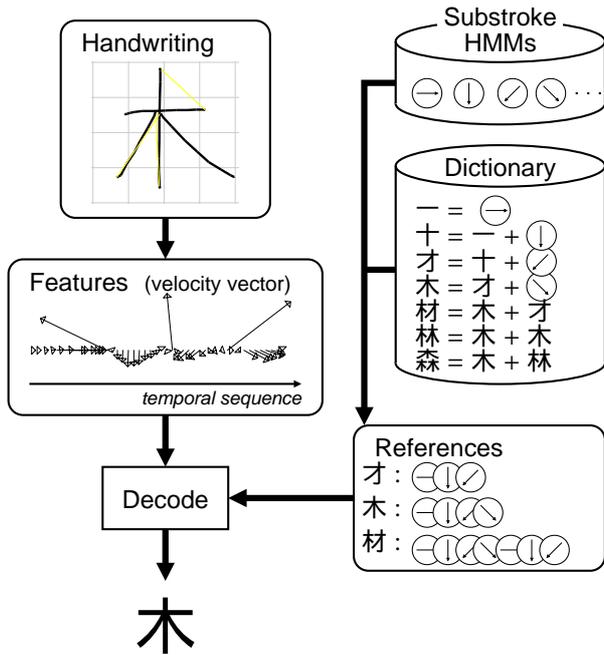


Figure 1. System configuration.

3. Untrained characters can be recognized by just adding their definitions of substroke sequence to the dictionary.
4. Characters written with different stroke orders can be recognized by using multiple definitions per one character in the dictionary.
5. Adaptation to the writer is performed with a few training characters. Since a single character is composed of a number of substrokes, sufficient adaptation data for all substrokes are provided from a few characters written.

All items above are discussed in this paper except for 4.

2. Kanji Handwriting Recognition Algorithm

2.1. Input Features

The proposed system basically consists of input features, HMMs, dictionaries and a decoder as shown in Fig.1. In this paper, we use pen positions (x, y) as the primary information although the input device (tablet) also provides other information such as pen pressure, pen tilts, etc. Let (dx, dy) be the feature vectors as the difference between two consecutive pen position samples. While the pen touches (pen down) the tablet surface, (dx, dy) represents the velocity vector of the pen positions sampled every certain period. When the pen leaves the tablet surface (pen up), (dx, dy) means a displacement vector between strokes before and

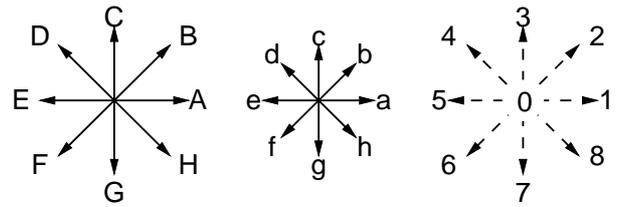


Figure 2. Substroke categories: A–H (a–h) are long (short) substrokes with pen down and 0–8 are the direction of pen up.

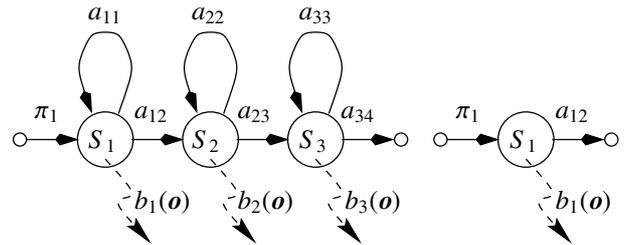


Figure 3. Substroke HMMs : (Left) pen down model, (Right) pen up model.

after pen-up as the pen position is not sampled while in the air.

2.2. Substroke HMMs

We model 25 substrokes of eight directions as shown in Fig.2; eight long strokes (A–H), eight short strokes (a–h), eight pen-up movement (1–8) and one pen-up-down movement (0). Eight categories are necessary and sufficient for representing Kanji characters. Though a continuous distribution is observed in written line directions, finer distinction gives no meaning in the classification sense. This is compared to the relationship between phonemes and allophones in speech recognition. The HMMs of these substrokes have a topology of left-to-right model shown in Fig.3. The pen-down models have three states for each that represent different stroke velocities, and pen-up models have only one state that outputs a displacement vector without self-loop probability. Here, let $\lambda^{(k)} = (\mathbf{A}^{(k)}, \mathbf{B}^{(k)}, \boldsymbol{\pi}^{(k)})$ be the set of HMM parameters of substroke k , in which

- $\mathbf{A}^{(k)} = \{a_{ij}^{(k)}\}$: The state-transition probability distributions from state S_i to S_j ,
- $\mathbf{B}^{(k)} = \{b_j^{(k)}(\mathbf{o})\}$: The probability distributions of observation symbols \mathbf{o} at state S_j ,
- $\boldsymbol{\pi}^{(k)} = \{\pi_i^{(k)}\}$: The initial state probability distributions.

The observation probability distribution is represented by a

Gaussian distribution given by

$$b_j(\mathbf{o}) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_j|}} \exp\left(-\frac{1}{2}(\mathbf{o} - \boldsymbol{\mu}_j)^t \Sigma_j^{-1} (\mathbf{o} - \boldsymbol{\mu}_j)\right),$$

with mean vector $\boldsymbol{\mu}$ and covariance matrix Σ . These model parameters can be trained by Viterbi training or Baum-Welch method.

2.3. Recognition

In this section, we describe a principle of decoding the time sequential feature of handwriting $\mathbf{O} = \mathbf{O}_1 \mathbf{O}_2 \cdots \mathbf{O}_T$, ($\mathbf{O}_t = (dx_t, dy_t)$) to the character defined by the substroke sequence $W^* = W_1^* W_2^* \cdots W_{N^*}$. The probability that the feature vector sequence \mathbf{O} is produced from HMM of stroke sequence $W = W_1 W_2 \cdots W_N$ is calculated by

$$P(\mathbf{O}|W) = \sum_{\text{all } q} P(\mathbf{O}, q|W),$$

$$P(\mathbf{O}, q|W) = \prod_{n=1}^N P(\mathbf{O}_{T_{n-1}+1} \cdots \mathbf{O}_{T_n}, q_{T_{n-1}+1} \cdots q_{T_n} | W_n)$$

$$= \prod_{n=1}^N \left(\pi_{q_{T_{n-1}+1}}^{(W_n)} \prod_{t=T_{n-1}+1}^{T_n} a_{q_{t-1}, q_t}^{(W_n)} b_{q_t}^{(W_n)}(\mathbf{O}_t) \right),$$

where $q = q_1 q_2 \cdots q_T$ is a state sequence that outputs \mathbf{O} , $T_0 = 0$ represents the start time, and T_N is equivalent to the time T . Therefore, the probability with which the observed sequence \mathbf{O} correspond to substroke sequence W can be calculated by using Bayes' Rule,

$$P(W|\mathbf{O}) = \frac{P(\mathbf{O}|W)P(W)}{P(\mathbf{O})}.$$

Since $P(\mathbf{O})$ is independent of W , the recognition result is

$$W^* = \operatorname{argmax}_W P(\mathbf{O}|W)P(W).$$

Practically, instead of calculating $P(\mathbf{O}|W)$, we often use the Viterbi algorithm that searches for state sequence q which gives maximum $P(\mathbf{O}, q|W)$.

2.4. Hierarchical Kanji Dictionary

The dictionary of Kanji characters is defined hierarchically as shown in Fig4. In this example, six Kanji characters “語”(word), “言”(say), “吾”(we), “五”(five), “口”(mouth), “一”(one) can be defined by 9 substroke models (A, F, a, g, h, 2, 3, 5, 6). The rule “語 = 言 2 吾” means that Kanji “語” is defined by combining “言” and “吾” with pen-up model “2”, where Kanji “言” and Kanji “吾” are treated as sub-Kanjis (radicals) of “語”. In our current dictionary, 6,353 Kanji characters are defined using approximately 6,700 rules. This hand-made dictionary may include wrong definitions and cause recognition errors. Automatic generation of hierarchical dictionary from data will be one of our future works.

W	=	語 言 吾 五 口 一
語	=	言 2 吾
言	=	、 6 - 6 - 短 6 - 短 6 口
吾	=	五 6 口
五	=	- 5 / 3 短 5 -
口	=	短 3 短 5 - 短
短	=	a g
-	=	A
-短	=	a
/	=	F
短	=	g
、	=	h

Figure 4. Hierarchical dictionary of six Kanji characters.

3. Experimental Evaluation

Handwriting database used in this evaluation is the JAIST IIPL (Japan Advanced Institute of Science and Technology, Intelligence Information Processing Laboratory) database that consists of two kinds of data sets; they are called α and β sets. The α set covers 1,016 Japanese characters of old and new educational Kanji containing 109,728 samples collected from 108 writers (0001 – 0108) with the standard stroke orders. This set is extremely useful in training substroke HMMs since simple Viterbi alignment can give correct stroke categories. The β set consists of 6,353 Kanji characters of JIS level 1 and 2; approximately 1,200 characters were collected from each of 80 people with free stroke orders.

3.1. Experiment 1: Substroke HMMs vs. Character HMMs

An experimental comparison between substroke HMM and character HMM was carried out on the 1,016 Japanese educational Kanji recognition task. The α set database was used in order to disregard a factor of writer specific difference of stroke order. The odd numbered 47 writers, i.e. 47,752 samples, were used for estimating the HMM parameters by Viterbi training method, and the remaining even numbered 49 writers were used for test. In case of character HMM, three states were assigned per stroke (not substroke) and one state was used per pen-up movement between strokes, therefore total $4N - 1$ states were used for a character composed of N strokes.

The recognition results are shown in Table 1, “1-best” means the correct recognition rate and “10-best” means the accumulative recognition rate of top 10 candidates. The recognition rate for training data sets is shown as “closed” and the recognition rate for evaluation writer sets is shown

Table 1. Comparison of recognition rates of the substroke HMM and the character HMM.

	1-best		10-best	
	open	closed	open	closed
Character HMM	95.92	(97.73)	99.84	(99.95)
Substroke HMM	95.34	(95.83)	99.47	(99.65)

Table 2. Recognition error samples from substroke HMM (S) and character HMM (C)

Substroke HMM > Character HMM: (S)–(C)			
input	(S)	(C)	error samples of (C)
問	100%	2%	問, 盟
貝	96%	10%	貝, 見
紀	100%	29%	絶, 続, 緑
白	90%	29%	自
固	100%	43%	個
Substroke HMM < Character HMM: (C)–(S)			
input	(S)	(C)	error samples of (S)
座	2%	100%	歴, 種, 笛, 宿
必	16%	98%	茶, 冬, 冷
宇	27%	100%	完, 序, 拳, 河
犬	29%	100%	太
人	8%	80%	八
5 lowest-performed characters: (S)+(C)			
input	(S)	(C)	error samples
夫	24%	37%	尺, 未, 失, 矢, 天
由	16%	49%	田
万	10%	59%	方
力	24%	51%	刀, 才 (S), 万 (C)
刀	35%	43%	力, 才 (S), 万 (C)

as “open”. These results show that there is no big differences between the recognition rate of substroke HMM and character HMM. In the case of substroke HMM, the difference of “closed” and “open” is small, thus substroke HMMs are trained by enough amount of data.

Two kinds of causes for recognition errors of substroke-based HMM were found and are shown in Table 2. One was the case that a simple dictionary can not discriminate substroke sequences (“必”, “犬”, “人”), the other was the case where the Kanji definition itself was wrong (“座”, “宇”) which seemed inevitable with manually crafted dictionary. In character-based HMMs, most error samples shared a same radical with input characters. In substroke HMMs, these characters are correctly recognized, because the decision of two Kanji characters “問” and “問” does not depend on the writing of common radical “門”, but depends only on the difference between “口” and “日”. Also, Kanji characters “貝” and “見” are distinguished by the difference of

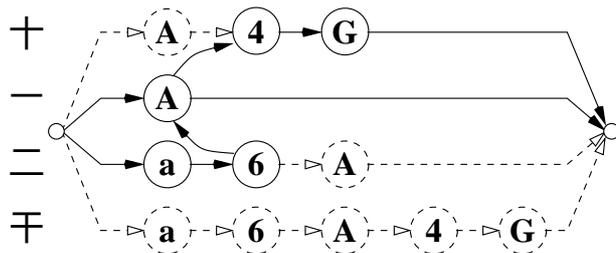


Figure 5. Substroke network representing four Kanji characters: “+ (A4G)”, “- (A)”, “二 (a6A)” and “干 (a6A4G)”.

“八” and “厶”.

Two significant advantages are found with substroke-based HMMs in comparison with whole character-based HMMs. First, the total memory size of models and dictionary is downsized to 1/40 with substroke HMMs. Let $(a_{ii}, \mu_{dx}, \mu_{dy}, \sigma_{(dx,dx)}, \sigma_{(dx,dy)}, \sigma_{(dy,dy)})$ be the state parameters of HMMs, about 24 bytes ((float) 4 byte \times 6) of memory are required per state. In case of 6,353 character HMMs, 7,387,560 bytes are required for 307,815 states (78,542 strokes). On the other hand, substroke HMMs require only 174,403 bytes, which is the total for 25 HMMs (1,368 bytes) and the Kanji dictionary (173,035 bytes). Second, the substroke HMM based decoder, which performs efficient network search, is nearly five times faster in case of full search (i.e., without hypothesis pruning). The decoder based on character HMM searches for a maximum of 307,815 states per input. In case of substroke HMM, we can cut search nodes down to 58,514 states by unifying common substroke subsequences in a network structure shown in Fig.5.

3.2. Experiment 2: Recognition of untrained Character

One advantage of substroke HMM is that any character can be recognized, if its definition is added to the dictionary, without having it in the training data. We carried out a 6,353 Japanese Kanji (JIS X0208-1983) recognition task, which was larger than experiment 1. The substroke HMMs trained in the previous experiment were used again in this experiment. This meant that 1,016 educational characters were included in the training data, while the remaining 5,337 were not. Data by 41 writers from the β set database were used for evaluation. Data from 19 writers were used as the writer-closed data, since data by the same writers extracted from the α were used for training. Other 22 writers were treated as “open writers”.

Compared with experiment 1, the recognition rate was lower because the task was more difficult and the evaluation data was written with free stroke order. The recognition

Table 3. Comparison of recognition rates of the trained characters and the untrained characters.

	1-best open (closed)	10-best open (closed)
trained characters	82.79 (81.92)	90.99 (89.66)
untrained characters	79.87 (78.54)	88.66 (87.51)

rate of untrained characters were lower than that of trained characters, though the difference was relatively small. One possible explanation of the performance difference between the trained and untrained is that untrained Kanji characters might include a wide variation in stroke orders which was unknown and was not described in the dictionary. Recognition accuracy of untrained characters may be improved by using multiple definitions per one character in the dictionary.

3.3. Experiment 3: Writer Adaptation

It is generally considered to be advantageous to use a writer-dependent HMM, if the writer is known to the decoder. It would take a long time, however, to prepare a large amount of training data from a specific writer. Therefore, it is desirable to adapt HMMs to the writer with a small amount of training data. We employ the maximum a posteriori (MAP) method [5] for writer adaptation. After adaptation, the mean vector of state S_j is given by

$$\hat{\mu}_j = \frac{N_j}{N_j + \tau} \bar{\mu}_j + \frac{\tau}{N_j + \tau} \mu_j,$$

where N_j denotes the number of adaptation data points to which state S_j is aligned, and $\bar{\mu}_j$ is the mean vector of those N_j frames. τ is a weighting factor of a priori knowledge to the adaptation data and we fixed it to 15.0. The data set used for experiments was α set and recognition task was to classify 1,016 Kanji vocabularies. Among them, 10 characters “一”(one), “右”(right), “雨”(rain), “円”(circle), “王”(king), “音”(sound), “下”(down), “火”(fire), “花”(flower), “貝”(shell) were used for adaptation and the other 1,006 characters were used for evaluation.

Table 4 shows the results of writer adaptation for about five writers whose recognition rate was lower than the other 44 writers before adaptation in experiment 1. We can see that the recognition rate after adaptations is improved and it is close to that of writer dependent HMM.

4. Conclusion

We have described our proposed on-line handwriting recognition method based on substroke HMMs and its base-

Table 4. Results of adaption to the 5 lowest-performing writers in Experiment 1.

Writer	Before adaptation	After adaptation	Writer dependent
	best-1 / ~10	best-1 / ~10	best-1 / ~10
0052	89.36 / 98.61	92.64 / 99.30	96.82 / 99.80
0072	91.55 / 99.01	92.94 / 99.01	94.63 / 99.60
0044	92.35 / 98.91	93.24 / 98.91	95.63 / 99.60
0056	92.45 / 99.20	94.04 / 99.30	96.12 / 99.70
0012	92.64 / 98.71	95.13 / 98.91	96.62 / 99.30

line performance. It has been shown that substroke HMMs are superior to whole character-based HMMs on the following three points: 1) the memory requirement for the dictionary and models is significantly small, 2) any characters defined in the dictionary can be recognized with limited training characters, and 3) adaptation to writer is easily performed with a few sample characters provided by the user.

References

- [1] J. Hu, M. K. Brown, and W. Turin. “HMM Based On-Line Handwriting Recognition”. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(10):1039–1045, Oct. 1996.
- [2] H. Itoh and M. Nakagawa. “An On-line Handwritten Character Recognition Method based on Hidden Markov Model” (in Japanese). *Technical report of IEICE*, PRMU97-85:95–100, July 1997.
- [3] R. Kohle and T. Matsumoto. “Pruning Algorithm for HMM On-line Handwriting Recognition” (in Japanese). *Technical Report of IEICE*, PRMU97-5:33–39, May 1997.
- [4] A. Kundu and P. Bahl. “Recognition of Handwritten Script: A Hidden Markov Model Based Approach”. *Proc. ICASSP '88*, 2:928–931, Apr. 1988.
- [5] C. H. Lee, C. H. Lin, and B. H. Juang. “A Study on Speaker Adaption of the Parameters of Continuous Density Hidden Markov Models.”. *IEEE Trans. Signal Processing*, 39(4), Apr. 1991.
- [6] R. Nag, K. H. Wong, and F. Fallside. “Script Recognition Using Hidden Markov Models”. *Proc. ICASSP '86*, 3:2071–2074, Apr. 1986.
- [7] T. Nonaka and S. Ozawa. “Online Handwritten HIRAGANA Recognition Using Hidden Markov Models” (in Japanese). *IEICE Trans. (D-II)*, J74-D-II(12):1810–1813, Dec. 1991.
- [8] T. Starner, J. Makhoul, R. Schwartz, and G. Chou. “On-Line Cursive Handwriting Recognition Using Speech Recognition Methods”. *Proc. ICASSP '94*, 5:125–128, Apr. 1994.
- [9] K. Takahashi, H. Yasuda, and T. Matsumoto. “On-line Handwritten Character Recognition Using Hidden Markov Model” (in Japanese). *Technical report of IEICE*, PRMU96-211:143–150, Mar. 1997.