On-Line Handwritten Kanji String Recognition Based on Grammar Description of Character Structures

Ikumi Ota, Ryo Yamamoto, Takuya Nishimoto and Shigeki Sagayama
The University of Tokyo.
{*ota, yamaryo, nishi, sagayama*}*@hil.t.u-tokyo.ac.jp*

Abstract

In this paper, we evaluate a method for on-line handwritten Kanji character recognition by describing the structure of Kanji using Stochastic Context-Free Grammar (SCFG), and extend it in order to recognize Kanji strings. In this method, we turn attention to the hierarchical structure of Kanji characters which consist of character-parts and strokes, and consider all character patterns or strings to be generated from SCFG with stochastic stroke shape and position relationship between character-parts. Describing Kanji with a few stroke shape and relative position labels, the method enables efficient training and thus robust recognition. We evaluated the recognition performance on several domains of Kanji, and on Kanji strings consist of 2 or 3 characters and gained the recognition rate of 99.29 – 97.40% for characters and 90.80% for strings.

1. Introduction

On-line recognition technology of handwritten Kanji characters (Chinese characters used in Japanese written language) and Kanji string is widely used these days and there is a growing demand for more accuracy. Different from alphabetical characters, Kanji are complicated in shape and huge in number, consist of many strokes located in specified positions, and have recursive structure. The structure of Kanji characters must be taken into account in order to improve recognition accuracy.

In on-line Kanji character recognition using the handwriting time sequential information, Nakai et al. [2] successfully applied Hidden Markov Models (HMM) to Kanji recognition with the same framework as phoneme-based continuous speech recognition, while HMM had been generally believed to be inappropriate to recognize Kanji due to their large variety and complexity, unlike alphanumeric characters. Since the method models Kanji as the sequence of substrokes shared with many Kanji, the model can be efficiently trained and the method showed great effect on stroke recognition accuracy. But the use of positional information is limited.

In order to take into account the relative position information, Kang and Kim [1] described the structure of Hangeul characters using a graph connecting character-parts according to positional dependency. Following [1], Tokuno et al. [5] used "Bayesian network" which connect every two character-parts that have a location dependency through a directional graph, and trained the relative position. The recognition performance using this method increases, but parameters need to be trained for each Kanji, thus large quantity of training data is required.

To solve this problem, we have proposed a new approach to handwritten Kanji recognition based on SCFG to efficiently model the layered structures inside Kanji. In [3], experimental results showed improvements in performance for relatively simple characters and efficiency in training.

Recognition of Kanji string is even more difficult since it requires the segmentation of characters. One of the difficulties is that the segmentation and recognition are not independent.

In this paper we evaluated the perormamce of this method with a developed grammar to improve accuracy and cover larger domain of Kanji and extended the method to recognize Kanji string. In section 2, we explain the details of the proposed method, and in section 3 we present its performance evaluation through recognition experiments.

2. Proposed method

2.1. Substroke-HMM Kanji recognition

We briefly describe here previous research using substroke-HMM to recognize Kanji, and shall refer to

978-1-4244-2175-6/08/$25.00 ©2008 IEEE

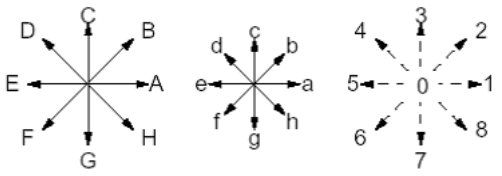

Figure 1. 25 substroke symbols representing 8-directional pen-down (long, short) and pen-up substrokes [2]

[2] for a more detailed presentation.

Since meaningful distinctions in substrokes (linear segments of strokes) exist in only 8 directions and 2 lengths that make distinguish Kanji categories as depicted in Figure 1, all written strokes in Kanji can be regarded as probabilistic deviation from typical shapes of 16 sufficient and necessary substroke symbols substroke symbols (A–H and a–h) without confusion. With additional 9 symbols for pen-up moves between strokes, all Kanji categories can be efficiently represented by "spellings" such as "G3a6A" for "上". This idea is parallel to that between phonemes and phonetical segments in phoneme-based continuous speech recognition framework where phonemes are discrete symbols stochastically related by HMM to phonetical segments of observable feature vector sequences.

Sampling the coordinates of the point of the handwriting track at regular intervals, and taking the difference between two consecutive coordinates leads to the definition of a velocity vector. These vectors are used as feature values inside an HMM. This idea was successful, as describing complicated Kanji strokes using simple substrokes enables the sharing of substroke models among different Kanji and thus facilitates the training process. On the other hand, as the position information can only be used through pen-up models, it is difficult to distinguish Kanji that have the same substroke representation but are different in stroke position, such as "八" and "人".

To solve this problem, Tokuno et al. [5] proposed a "Bayesian Network" model, which connects every two character-parts which have location dependency inside a directional graph, and trained the relative position. This method improved the previous one, but as we mentioned earlier, it needs a certain amount of data for each Kanji to be recognized, and cannot recognize Kanji for which it has not been trained.

2.2 Handwritten Kanji model with CFG

To represent relative positions between Kanji components with a small number of models shared among different Kanji to simplify the training, we pay attention to the hierarchical structure of Kanji and introduce a stochastic context-free grammar (SCFG) by combining strokes into Kanji. We model the stochastic generation of handwritten Kanji string with the SCFG like shown in Figure 2.

Generation rules of non-terminal symbols	Character generation rules	S → K_{str} K_{str} → K_{str} (next) K K_{str} → K K → 山 | 川 | 明 | …
	Character-part generation rules	明 → 日 (right) 月 山 → 丨 (under) 凵 凵 → 𠂆 (right) 丨
Generation rules of terminal symbols		𠂆 → 乚 丨 → 丿

Figure 2. Examples of the handwritten Kanji string generation rules. S, K_{str}, K**, are the start symbol, the symbol of Kanji string, and that of Kanji character respectively.**

The grammar consists of generation rules of the non-terminal symbols: the characters and the character-parts, and generation rules of the terminal symbol: handwritten strokes. This is different from the normal SCFG which generates sequences of words in the following three points.

1. The terminal symbols are the observed handwritten stroke shapes. Generation rules of the terminal symbols are written in the form of $r_t = \langle A \rightarrow a \rangle$, where A is a character-part symbol of a stroke, and a is a handwritten stroke.

2. The non-terminal symbol A has its position b_A. In this paper, b_A is the bounding box of the observed handwritten strokes corresponding to the character or character-part A, or is the bounding box of the last character of A when A is Kanji string K_{str}.

3. Generation rules of the non-terminal symbols are written in the form of $r_n = \langle A \rightarrow B(\mathrm{op})C \rangle$.

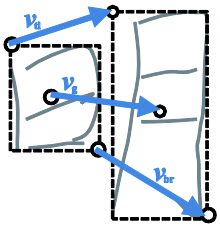

Figure 3. Feature vector used in the placement model.

A, B, and C are the symbols of characters or character-parts, and (op) is the "placement operator" like (next), (right), or (under), meaning that the symbol C should be placed next to, in the right position of, or in the lower position of B.

2.3. Stochastic generation of handwritten Kanji string

The generation rules are stochastically applied. When a non-terminal symbol generation rule $r_n = \langle A \rightarrow B(\mathrm{op})C \rangle$ is applied, its probability $p(r_n)$ is expressed as

$$p(r_n) = p_{r_n}^{\mathrm{rew}} \cdot p_{r_n}^{\mathrm{pos}}, \tag{1}$$

where $p_{r_n}^{\mathrm{rew}}$ is the rewriting probability, showing the frequency that the rule r_n is applied to the symbol A, and $p_{r_n}^{\mathrm{pos}}$ is the positioning probability. The positioning probability is a function of (op), b_B, and b_C, modeling the stochastic positioning of the generated symbols according to the placement operator (op) of r_n.

We assumed in this paper the rewriting probability is flat, and we modeled the positioning probability using the normal distribution of the feature vector $(l_{\mathrm{tl}}, l_{\mathrm{br}}, l_{\mathrm{g}}, \theta_{\mathrm{tl}}, \theta_{\mathrm{br}}, \theta_{\mathrm{g}})$ representing relative position of b_B and b_C, shown in Figure 3, where $(l_{\mathrm{tl}}, \theta_{\mathrm{tl}})$ is v_{tl} represented in polar coordinates. we used diagonal covariance matrix here.

The application probability $p(r_t)$ of a terminal symbol generation rule r_t is

$$p(r_t) = p_{r_t}^{\mathrm{rew}} \cdot p_{r_t}^{\mathrm{shape}}, \tag{2}$$

where $p_{r_t}^{\mathrm{rew}}$ is the rewriting probability, and $p_{r_t}^{\mathrm{shape}}$ is the shape probability that models the stochastic generation of the handwritten stroke shape. We modeled the stroke shape probability with the Substroke-HMM [2].

2.4. Kanji string recognition

Kanji string recognition problem is to find the maximum posterior probability derivation

$$Y = \{r_{n1}, r_{n2}, \ldots, r_{nN}, r_{t1}, r_{t2}, \ldots, r_{tT}, \} \tag{3}$$

for the observed stroke sequence X.

$$\hat{Y} = \underset{Y}{\mathrm{argmax}}\, P(Y|X) \tag{4}$$

$$= \underset{Y}{\mathrm{argmax}} \prod_{i=1}^{N} p_{r_n i}^{\mathrm{rew}} \prod_{i=1}^{N} p_{r_n i}^{\mathrm{pos}} \prod_{i=1}^{T} p_{r_t i}^{\mathrm{rew}} \prod_{i=1}^{T} p_{r_t i}^{\mathrm{shape}} \tag{5}$$

This problem can be solved with parsing. In this paper we used CYK parsing algorithm. For more detailed information on this parsing process, see [3].

Efficient training is expected since this method, making use of hierarchical combination of primitive shape and placements, models Kanji characters with a small number of parameters. And the robust recognition is also expected for this method optimizes the shape and structure simultaneously, and free from any intermediate hard-decision of segments.

3. Evaluation Experiments

3.1. Baseline performance evaluation

To evaluate the performance of this method, we conducted three evaluation experiments. The first one is to evaluate the baseline performance and the training efficiency.

In this paper, we used JAIST-IIPL on-line character database shown in Figure 4 divided into four subdatasets like shown in Table 1. And we prepared the corresponding four grammar dictionaries for handwritten Kanji shown in Table 1 together. Dic-1 covers the old/new educational Kanji written with fixed stroke order, which is revised based on the dictionary we used in the previous work [3] to improve accuracy. Dic-1 includes 17 shapes and 18 placements, while the dictionary used in the previous work includes 21 and 12.

We conducted the isolated Kanji character recognition experiments using the dataset of Set-1, 1,016 old/new Japanese educational Kanji characters written in the fixed writing order by 97 writers, and the corresponding handwritten grammar dictionary of Dic-1.

We used the data of 10 writers for training, those of the same 10 writers for CLOSED evaluation, those of other 30 writers for OPEN evaluation.

Table 1. Datasets and their corresponding grammars

Kanji Category	Dataset			Corresponding Grammar			
	Name	stroke order	source	Name	# of rules	# of shapes	# of placements
old/new educational Kanji (1,016 characters)	Set-1	fixed	α-set	Dic-1	1,885	17	18
	Set-2	free	β-set	Dic-2	2,507	34	18
JIS level-1 (2,965)	Set-3	free	β-set	Dic-3	5,305	34	18
JIS level-2 (6,353)	Set-4	free	β-set	Dic-4	9,610	34	18

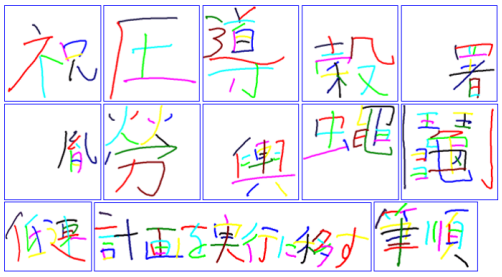

Figure 4. Samples in JAIST-IIPL Database

Table 2. Baseline performance

	N-th accumulative recognition rate(%)				
	1	2	3	5	10
CLOSED	99.42	99.91	99.95	99.97	99.97
OPEN	99.34	99.82	99.88	99.90	99.90

The result of accumulative recognition accuracy rate is shown in Table 2. The rate, 99.42% for CLOSED and 99.34% for OPEN, is obviously better than that in the previous work [3], 98.12% and 97.72%. And the fact that the performances of this method in OPEN and CLOSED conditions are very close to each other shows that our method achieved the efficient training and requires small amount of training data.

3.2. Performances on various domains of Kanji

The second experiment is to see how the performance changes along with the expansion of the target domain of Kanji.

Using the same training data as used in 3.1, we changed the dictionary and evaluation dataset. We used the grammar of Dic-1 to Dic-4 and the dataset of Set-1 to Set-4 written by 20 writers.

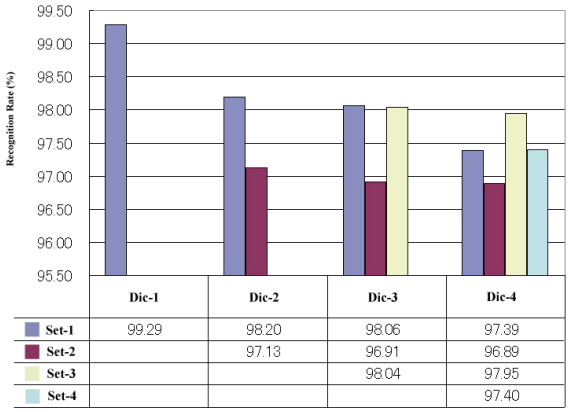

Figure 5. Performances on various domains of Kanji

The recognition accuracy rate is shown in Figure 5. Comparing the results of (Dic-1, Set-1), (Dic-2, Set-2), (Dic-3, Set-3), and (Dic-4, Set-4), we can see that the performance stays relatively high with the expansion of the target domain. And the performances on each dataset with the expanding domains of dictionary do not decrease a lot. So we can see that this method is robust against the change of the domain and requires only the replacement of the dictionary, not necessarily training using larger dataset, to adapt the larger target domain of Kanji.

3.3. Kanji string recognition

The third evaluation is on Kanji string recognition. We used for evaluation handwritten phrases consist of 2 or 3 old/new educational Kanji characters written by 20 writers in γ-set of the database, removing data with errors. We used Dic-2 for the grammar of Kanji characters.

The string recognition accuracy rate is 90.80%, while that of the method proposed in [4], which uses the

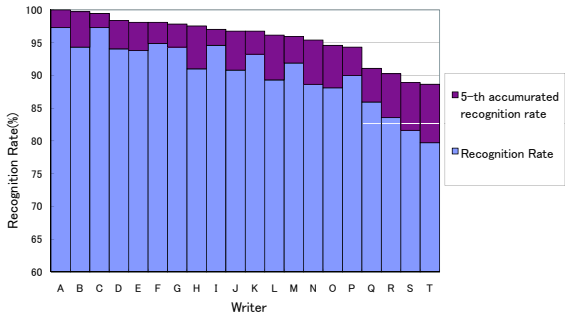

Figure 6. String recognition rate for each writer

Table 3. The examples of mis-recognized Kanji strings

Answer		Mis-recognized results	Error rate(%)
向上	→	向土, 局土, 白上	90
工学	→	土学	75
任意	→	件意, 弁意	50
感謝	→	感言射, 感言身子	25
線分	→	糸泉分, 線欠	20
不注意	→	友注意, 木注意, 不玉退八	25
観測	→	声責測, 考現測, 勧科利	20
音節	→	音建刀, 音算, 立由節, 専節	30
一人	→	大	75
一方	→	切	30
中心	→	忠, 中火	15
帰納	→	帰糸分, 白虫納, 帰糸分	20
長短	→	長矢豆	10

Substroke-HMM with pen-up model to recognize Kanji strings, is 81.12%. We could see the effectiveness of the proposed method here.

The accuracy rate by writer is shown in Figure 6. The performances for the several particular writers, the writer Q to T, are much worse than others.

The examples of mis-recognized data is shown in Table 3. The main reason for mis-recognition seems to be classified in some categories. (1) because the stroke order is not supported in the dictionary, (2) because relative position between character-parts within one character is mis-recognized, and (3) because relative position between characters is mis-recognized.

Better design of the dictionary is important to adapt various writing styles and improve performance. And this design is better to be done automatically by computer using dataset in the future.

4. Conclusion

We evaluated a method for on-line handwritten Kanji character recognition using SCFG, and its extension for Kanji string recognition. The method considers all character patterns or strings to be generated from SCFG with stochastic stroke shape and position relationship between character-parts. Describing Kanji with a few stroke shape and relative position labels, the method enables efficient training and thus robust recognition. We evaluated the recognition performance on several domains of Kanji characters, and on Kanji strings consist of 2 or 3 characters, and gained the recognition rate of 99.29 – 97.40% for characters and 90.80% for strings. Our future work includes the application of this method to other class of characters like Hiragana, Katakana characters, Alphabets and Digits, automatic design of the dictionary, and adaptation to other character placement like vertical writing and the writing with line breaks.

Acknowledgment

This research is done on the basis of the one at JAIST-IIPL (`http://iipl.jaist.ac.jp/`) and its on-line handwriting database. We thank the JAIST-IIPL members.

References

[1] K. Kang and J. Kim. Handwritten Hangul Character Recognition with Hierarchical Stochastic Character Representation. *Proc. of ICDAR'03*, pages 212–216, August 2003.

[2] M. Nakai, N. Akira, H. Shimodaira, and S. Sagayama. Sub-stroke Approach to HMM-based On-line Kanji Handwriting Recognition. *Proc. of ICDAR*, pages 491–495, Sep. 2001.

[3] I. Ota, R. Yamamoto, T. Nishimoto, and S. Sagayama. Online Handwritten Kanji Recognition on Inter-stroke Grammar. *Proc. of ICDAR*, 2:1188–1192, Sept. 2007.

[4] T. Sudo. Research on On-Line Handwritten String Recognition Based on Hidden Markov Model (in Japanese). *Master's thesis, JAIST*, Feb. 2002.

[5] J. Tokuno, M. Nakai, H. Shimodaira, S. Sagayama, and M. Nakagawa. On-line Handwritten Character Recognition Selectively employing Hierarchical Spatial Relationships among Subpatterns. *Proc. of IWFHR*, Oct. 2006.