Online Handwritten Kanji Recognition Based on Inter-stroke Grammar

Ikumi OTA, Ryo YAMAMOTO, Shinji SAKO[†] *and Shigeki SAGAYAMA*
Graduate School of Information Science and Technology
The University of Tokyo
{ota,yamaryo,sako,sagayama}@hil.t.u-tokyo.ac.jp

Abstract

This paper presents a new approach to online recognition of handwritten Kanji characters focusing on their hierarchical structure. Stochastic context-free grammar (SCFG) is introduced to represent the Kanji character generating process in combination with Hidden Markov Models (HMM) representing Kanji substrokes and to improve the recognition accuracy of important and frequently used Kanji characters in which inter-stroke relative positions play important roles. Combining the stroke likelihood and the relative-position likelihood between character-parts in the parsing process is expected to compensate their ambiguities. By modeling relative positions and share the models across distinct Kanji categories, a small training data can yield effective results and enables us to recognize Kanji simply by defining the SCFG rules to represent their structures without training data. Experimental results on an online handwritten Kanji database from JAIST (Japan Advanced Institute of Science and Technology) showed significant improvements in the recognition rates of some important Kanji with relatively fewer strokes and also showed little difference between the trained- and the non-trained Kanji in recognition rates.

1 Introduction

Online recognition technology of handwritten Kanji (Chinese characters commonly used in Japanese language) is now widely used in pen-input interfaces such as PDAs or handy phone devices, and is expected to become more popular and to see the range of its applications broaden in the future. However, the accuracy of automatic recognition of cursive writings is far from that of human. Kanji are indeed combinations of strokes located in specified positions, and, even in the cursive written characters, the shape of the strokes and the relative positions between strokes or character-parts are preserved to some extent. The structure of Kanji characters must be taken into account in order to improve recognition accuracy.

In online 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. This method showed great effect on stroke recognition accuracy indeed, 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 Hangul characters using a graph connecting character-parts according to positional dependency. Following [1], Tokuno et al. [3] 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 all Kanji, thus large quantity of training data is required.

To solve this problem, we choose in this article to express the position relationships using a limited number of labels. However, determining which label each character-part belongs to only by its coordinates might not be efficient, as the way of writing character-parts varies with the writer. In order to take this matter into account, it is necessary to consider the recognition of the character-part each stroke combination forms as a stochastic problem. Such a problem can be effectively handled using SCFG, according to research on mathematical expression recognition by Yamamoto et al. [4]. We propose a new approach to handwritten Kanji recognition based on SCFG, and consider an efficient way to use layered structures inside Kanji. In the following, after related previous work is briefly presented, SCFG and its application to our method will be introduced. Then, modeling relative positions between strokes/character-parts, the stroke model likelihood calculation process, and finally the SCFG parsing procedure will be given. Experimental evaluation of the method will be presented in Section 3, followed by conclusion and future research.

[†] currently with Graduate School of Engineering, Nagoya Institute of Technology, Nagoya, Japan.

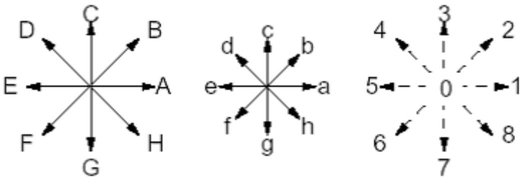


Figure 1. 25 substroke symbols representing 8-directional pen-down (long, short) and pen-up substrokes [2]

2 Kanji Recognition based on Inter-stroke Grammar

2.1 Substroke-HMM Kanji recognition

We briefly describe previous research using substroke-HMM to recognize Kanji, and shall refer to [2] for a more detailed presentation.

Since meaningful distinctions in substrokes (linear segments of strokes) exist in only 8 directions and 2 lengths depicted in Fig. 1 that distinguish Kanji categories, all written strokes in Kanji can be regarded as probabilistic deviation from typical shapes of 16 sufficient and necessary 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”, e.g., “G3a6A” for “上”. This idea was first motivated by relationship between phonemes and phonetical segments in the phoneme-based continuous speech recognition framework where phonemes are discrete symbols stochastically probabilistically related by Hidden Markov Models 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. [3] 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 was efficient in handling the relative positions, 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.

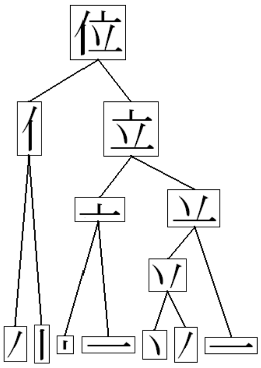


Figure 2. An example of hierarchical structure of Kanji

2.2 Hierarchical structure in Kanji

To represent relative positions between Kanji components with a few number of models shared among different Kanji and 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.

Kanji are composed of some character-parts, i.e., combination of two or more strokes. For example, the Kanji “位” displayed in Fig. 2 consists of character-parts “亻” and “立”, themselves composed of smaller character-parts which can finally be divided into strokes. Kanji have a certain number of classes of hierarchical structure inside themselves, with which Kanji characters may be represented better.

2.3 Stochastic context-free grammar

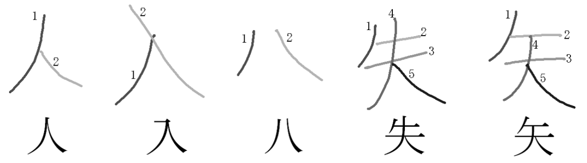
Context-free grammar is a suitable method for producing or analyzing any self-reflexive language, which can be described by (V_N, V_T, P, S), where V_T is a set of terminal symbols, which cannot be extracted, V_N a set of non-terminal symbols, S a set of start symbols, and P a set of production rules which can be expressed in the following form:

$$A \to \alpha \qquad (A \in V_N,\ \alpha \in (V_N \cup V_T)^*).$$

All production rules can actually be transformed to either one of the following rules, the so-called “Chomsky normal form”:

$$\begin{array}{lcll} A & \to & BC & (A, B, C \in V_N) \\ A & \to & a & (A \in V_N,\ a \in V_T). \end{array}$$

In this work, each handwritten stroke is considered as a terminal symbol, each stroke candidate and character-part as a non-terminal symbol, and each Kanji candidate as a start symbol. Kanji is analyzed by applying predefined production rules.



(The three Kanji “人”, “入” and “八” cannot be distinguished without using position information. In the same way, “矢” and “失” only differ in the penetration or not of the 4th stroke in the 2nd stroke.)

Figure 3. Examples of Kanji with same stroke representations

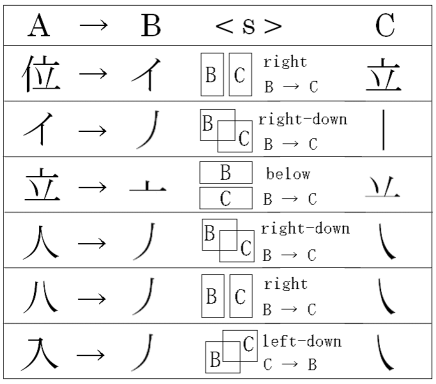
The information on positions between strokes or character-parts is also essential for high-accuracy recognition. Fig. 3 shows some examples of pairs of Kanji that are hard to distinguish from each other without considering relative position information. For example, “人” and “八” have the same stroke shapes and only the relative positions of the strokes are different. Furthermore, in practical writing, it is hard to declare definitely whether the second stroke is located in the “right” or “right-up” position with regard to the first stroke. Therefore, it is necessary to perform decisions on the relative position stochastically.

Our SCFG was created in the following way: the production rules for non-terminal symbols are of the form $p =< A \rightarrow BC,\ s >$, where A, B, C represent either Kanji themselves or character-parts, s describes the relative position between B and C, and the probability of applying the rule corresponds to the likelihood of B and C being located according to the relation s (position likelihood). The production rules to generate terminal symbols are of the form $q =< t \rightarrow \alpha >$, where t and α represent the stroke shape and the handwritten stroke candidates, respectively, and the probability of applying the rule corresponds to the likelihood of α following the stroke model t (stroke likelihood). Here, let $p_n =< A_n \rightarrow B_n C_n,\ s_n >$ be the n-th applied non-terminal symbol production rule, $q_m =< t_m \rightarrow \alpha_m >$ be the m-th applied terminal symbol production rule, and N and M be the number of non-terminal and terminal symbol generalizing rules applied, respectively. Then, the Kanji recognition problem can be described through the following mathematical expression:

$$X_0 = \operatorname*{argmax}_{X \in E_X} \prod_{n=1}^{N} P(p_n) \prod_{m=1}^{M} P(q_m). \tag{1}$$

In the expression above, X represents a hypothesis, E_X a set of hypotheses which can be generated from the grammar, and X_0 the most plausible hypothesis. By observing the equation above, Kanji recognition problem based on SCFG corresponds to the problem of finding the derivation which maximizes the product of stroke likelihood and position likelihood.

Unlike substroke-HMM used in [2, 3] to recognize complete Kanji, we use it here for likelihood calculation for the strokes. By expressing each stroke model as a concatenation of pen-down substrokes, likelihood of handwritten strokes can be calculated through the substroke-HMM framework as HMM gives a likelihood for each of stroke model candidates. The stroke likelihood and the relative-position likelihood between character-parts are expected to mutually compensate their ambiguities in the parsing process of the SCFG.



A	→	B	< s >	C
位	→	イ	right B → C	立
イ	→	ノ	right-down B → C	丨
立	→	亠	below B → C	䒑
人	→	ノ	right-down B → C	㇏
八	→	ノ	right B → C	㇏
入	→	ノ	left-down C → B	㇏

Figure 4. Example of entry of the SCFG lexicon

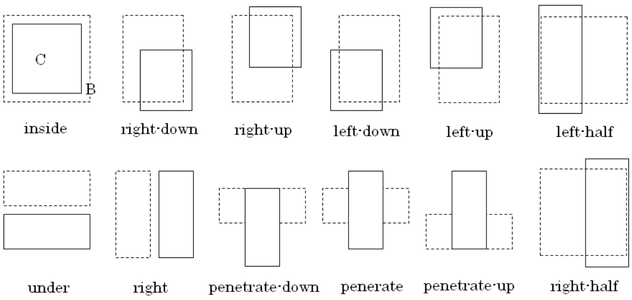


Figure 5. Relative position labels

2.4 Relative position between character-parts

We limited the position labels to the ones presented in Fig. 5 and assumed that all positions can be represented using these labels. In the following, we assume that every character-part X lies in a rectangular bounding-box which can be represented as $b_X \equiv (b_X^l, b_X^t, b_X^r, b_X^b)$ where $b_X^l, b_X^t, b_X^r, b_X^b$ describe the left, top, right, and bottom coordinate of the rectangle, respectively. Without loss of generality, one can suppose that the handwritten Kanji W considered lies in a unit square bounding-box. This corresponds to performing the following linear transformation on each bounding-box b_A, leading to b'_A:

$$\begin{aligned} b_W &= (b^l_X, b^t_X, b^r_X, b^b_X) \Rightarrow b^{'}_W \equiv (0,0,1,1) \\ b_A &= (b^l_A, b^t_A, b^r_A, b^b_A) \Rightarrow b^{'}_A \equiv (b^{l'}_A, b^{t'}_A, b^{r'}_A, b^{b'}_A) \\ &= \left(\frac{b^l_A - b^l_X}{b^r_X - b^l_X}, \frac{b^t_A - b^t_X}{b^b_X - b^t_X}, \frac{b^r_A - b^r_X}{b^r_X - b^l_X}, \frac{b^b_A - b^b_X}{b^b_X - b^t_X} \right). \end{aligned}$$

During training, we regard the difference between the converted bounding-boxes of the two character-parts (B and C) as the training feature value:

$$\begin{aligned} \Delta h_{BC} &= (\Delta b^l_{BC}, \Delta b^t_{BC}, \Delta b^r_{BC}, \Delta b^b_{BC}) \\ &= (b^{l'}_C - b^{l'}_B, b^{t'}_C - b^{t'}_B, b^{r'}_C - b^{r'}_B, b^{b'}_C - b^{b'}_B). \end{aligned}$$

Assuming that the feature in each position label s follows a normal distribution, the position likelihood $P_{pos}(B, C, s)$ of B and C being in positional relationship s can be written as follows:

$$P_{pos}(B,C,s) = \frac{1}{\sqrt{(2\pi)^4 \, |\Sigma_s|}} e^{-\frac{1}{2}(\Delta h_{BC} - \mu_s)^T \Sigma_s^{-1} (\Delta h_{BC} - \mu_s)},$$

where μ_s is the mean and Σ_s the covariant matrix of the distribution.

2.5 Parsing mechanism

To parse an SCFG based language, we modified the CYK (Cocke-Younger-Kasami) algorithm so as to handle with relative positions between parts-of-Kanji as follows:

Step 1 : Calculate the stroke likelihood for each input stroke with the Viterbi algorithm on stroke HMMs, and store it in the CYK matrix component $(1, j)_{j=1,2,...,L}$, where L is the number of rows of the CYK matrix and also corresponds to the number of strokes in the handwritten Kanji.

Step 2 : Search all production rules which can be applied to components $(1, j)$ and $(1, j+1)$, for $j = 1, \ldots, L-1$. If rule $p =< A \rightarrow BC, \ s >$ matches, calculate the product of the stroke likelihoods of the two strokes and the position likelihood. The result is the likelihood that the combination of the two strokes B and C is a character-part A. Thus, store A and the corresponding likelihood in the component $(2, j)$.

Step 3 : For components $(i, j)_{i=2,\cdots,L; j=1,\cdots,L-i}$, search all production rules applicable to components (i, j) and $(i, j+1)$, calculate the product of the likelihoods of the two character-parts and the position likelihood, and store it in $(i+1, j)$.

In this way, we finally get the recognition result stored in component $(L, 1)$. There might be multiple candidates, but we choose the one which belongs to the set of start symbols S and which has the higher likelihood as the result. To avoid underflow caused by very small likelihood values during the process above, logarithmic likelihood is actually taken and the product of the likelihoods is calculated as follows:

$$\log P(A) = \log P(B) + \log P(C) + W \log P_{pos}(B, C, s),$$

where $P(X)$ denotes the likelihood to be a character-part/stroke/Kanji X, and W is a weight value. We used $W = 5$ in the experiments shown in the next section.

3 Experimental evaluation

3.1 Database and grammar used in the experiment

Our method was evaluated using JAIST-IIPL (Japan Advanced Institute of Science and Technology, Intelligent Information Processing Laboratory) online handwritten character database from which the following two datasets were used:

α**-set**: Contains 1,016 educational Kanji for each writer written with correct order and number of strokes.

β**-set**: Contains 2,965 JIS Level-1 Kanji and 3,390 Level-2 Kanji. Some of them are written with a wrong order or number of strokes. We only used Level-1 Kanji in the experiment.

The Kanji lexicon used in the experiment contained 4,028 production rules, which supported the 2,965 Level-1 Kanji, including 1,016 educational Kanji, used in recognition, written in correct order. 12 position models and 21 stroke models were used to express the rules.

3.2 Evaluation and discussion

The evaluation was performed as follows. The likelihood for each start symbol was calculated, and the start symbol with the highest likelihood was chosen as the recognition result. For each writer, the error rate was calculated according to:

$$\left(1 - \frac{\text{Number of Kanji recognized correctly}}{\text{Number of Kanji used in recognition}} \right) \times 100\,(\%).$$

3.2.1 Writer independency

The position feature was trained using the α-set data of 10 writers for 1,016 Kanji, and evaluated using the same data sets of 10 writers used for training (CLOSED) and the data sets of 50 writers not used for training (OPEN). The recognition error rate is displayed in Table 1. The difference in average rate between CLOSED and OPEN is actually rather small, which shows that this method does not strongly depend on writers and works effectively as a character recognition system.

Table 1. Recognition results on closed and open data sets

Category	Error rate	Range of rate
CLOSED	1.88%	0.96% – 3.05%
OPEN	2.28%	1.48% – 3.35%

Table 2. Examples of characters with improved recognition results

	Error rates and error samples			
Kanji	Proposed		Previous	
反	0%	-	40%	友
失	6%	矢	54%	矢, 史
人	0%	-	74%	八
入	24%	八	84%	八, 人

Other samples with improved recognition results:
刀/力, 犬/太, 売/壱, 午/牛, etc.

3.2.2 Comparison with previous work

By using the position relationship between character-parts effectively, some pairs of Kanji are now recognized well which used to be difficult to distinguish using Substroke-HMM [2]. Table 2 and 3 show some examples of Kanji whose recognition rate has been improved or deteriorated compared with the previous work. Kanji pairs whose substroke descriptions are the same or similar, such as "人/入/八" or "矢/失", can now be distinguished well. This indicates that the proposed method, considering inter-stroke position relationships, is effective at distinguishing simple Kanji, especially ones which consist of few strokes. On the other hand, recognition of Kanji which have inter-stroke relationships which cannot be well described using the 12 position models deteriorated. It seems necessary to use more position models and to re-write the grammar descriptions for some Kanji to reflect the position relationships more effectively and make the method work better.

3.2.3 Character independency in training

1,014 production rules were added to the Kanji lexicon to tackle the problem of characters written in wrong order. this lexicon was applied to the β-set and performed recognition of the Level-1 Kanji for 20 writers. Position features were trained using the same 10 writers from the α-set as in the experiment mentioned above. Finally, recognition was performed on 2,965 Kanji, for a total of 11,618 handwritten characters, and the error rate only increased by 0.43% to 2.71%. This shows that even if the number of production rules is increased, the level of recognition rate using this method is expected to not strongly deteriorate. This means that this method does not need training data for all the Kanji to be recognized, but just to write the corresponding production rule on the lexicon.

Table 3. Examples of characters with deteriorated recognition results

	Error rates and error samples			
Kanji	Proposed		Previous	
史	58%	央	34%	央, 兄
月	52%	日, 円	14%	日

Other samples with deteriorated recognition results:
四/皿/冊, 下/寸, etc.

4 Conclusion

In this paper, a unified framework of Kanji recognition based on SCFG combining stroke and position likelihoods has been presented. Experimental results show that this framework improves the performance for relatively simple and thus frequently used characters, and that it leads higher recognition performance with a limited amount of training data, and on characters not included in the training data. Future research includes extension to more than 6,500 Kanji categories (and even all Chinese characters), selection of better relative position feature to more flexibly treat the relative position between Kanji components, automatic creation of an optimal grammar description for each Kanji to reflect most the inter-stroke position relationship, and extension to simultaneous segmentation and recognition of multiple character input in the SCFG framework.

5 Acknowledgments

The authors would like to express special thanks to Dr. M. Nakai, Toyama Prefectural University, for his valuable assistances and providing his detailed experimental results. We also thank other former members of JAIST-IIPL (`http://iipl.jaist.ac.jp/`) for their effort in collection of JAIST-IIPL handwriting database used in this work.

References

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

[2] M. Nakai, N. Akira, H. Shimodaira, and S. Sagayama. Substroke to HMM-based On-line Kanji Handwriting Recoognition. *Proc. of ICDAR'01*, pages 491–495, Sep. 2001.

[3] J. Tokuno, M. Nakai, H. Shimodaira, and S. Sagayama. Online Handwriting Recognition Selecting Hierarchial Spatial Relationship of Subcharacters (in Japanese). *Technical Report of IEICE, PRMU*, 79(7):95–100, Sep. 2005.

[4] R. Yamamoto, S. Sako, T. Nishimoto, and S. Sagayama. On-Line Recognition of Handwritten Mathematical Expression Based on Stroke-Based Stochastic Contexgt-Free Grammar. *Proc. of 10th IWFHR*, pages 491–495, Oct. 2006.