

# On-line Overlaid-Handwriting Recognition Based on Substroke HMMs

Hiroshi SHIMODAIRA\*<sup>1</sup>, Takashi SUDO\*<sup>2</sup>, Mitsuru NAKAI\*<sup>1</sup>, and Shigeki SAGAYAMA\*<sup>3</sup>

\*<sup>1</sup> Graduate School of Information Science, Japan Advanced Institute of Science and Technology  
{sim,mit}@jaist.ac.jp

\*<sup>2</sup> Core Technology Center,

TOSHIBA CORPORATION Digital Media Network Company

takashi2.sudo@toshiba.co.jp

\*<sup>3</sup> Graduate School of Information Science and Technology,

The University of Tokyo

sagayama@hil.t.u-tokyo.ac.jp

## Abstract

This paper proposes a novel handwriting recognition interface for wearable computing where users write characters continuously without pauses on a small single writing box. Since characters are written on the same writing area, they are overlaid with each other. Therefore the task is regarded as a special case of the continuous character recognition problem. In contrast to the conventional continuous character recognition problem, location information of strokes does not help very much in the proposed framework. To tackle the problem, substroke based hidden Markov models (HMMs) and a stochastic bigram language model are employed. Preliminary experiments were carried out on a dataset of 578 handwriting sequences with a character bigram consisting of 1,016 Japanese educational Kanji and 71 Hiragana characters. The proposed method demonstrated promising performance with 69.2% of handwriting sequences being correctly recognized when different stroke order was permitted, and the rate was improved up to 88.0% when characters were written with fixed stroke order.

## 1. Introduction

As mobile and wearable computing devices have become popular, a number of character input interfaces have been developed. The keyboard interface is mainly used in notebook personal computers (PCs), while an on-line handwriting interface is preferable to the keyboard interface in personal digital assistants (PDAs). There have been several handwriting input methods proposed for European alphabets, for example, single-box writing interface for unistroke characters [2], multi-box writing interface for multi-stroke characters, Treadmill interface [8], and write-anywhere interface [10]. For Japanese and Chinese on-line character recognition, it is common to employ the multi-box writ-



**Figure 1. Examples of overlaid handwritings.**  
The numbers in the brackets denote the numbers of characters contained in the image.

ing interface, where users write each character in each box one after another. The merit of employing the multi-box interface is that character segmentation is not needed and as a result, isolated-character recognition techniques can be applied rather than using continuous character recognition techniques [6]. Since most of the Japanese and Chinese characters are multi-stroke, the multi-box writing interface is a good compromise between the character recognition accuracies and usability.

If we imagine much smaller devices than PDAs, such as cellular phones or a small touch-pad attached to human bodies for wearable computing, the multi-box writing interface is no longer of use due to the very limited writing space. Since the writing area is small enough to write only a single character at once, it is identical to the single-box writing interface. Furthermore, in order to achieve a user-friendly interface, it would be preferable that users need not provide the interface with information of end of character explicitly by a pause or a time out between entries. Assuming that users write characters continuously without pauses on a single-box writing area, images of the input characters would be overlaid with each other as are shown in Fig. 1. We call the handwritings written in this style 'overlaid handwritings'.

Problems arise when the overlaid-handwriting interface is designed. For example, (i) the segmentation problem, and (ii) the echo back problem. The segmentation problem is that the boundaries between characters are not explicitly given, which forces the system to devise a special function of estimating the character boundaries explicitly or implicitly. The echo back problem occurs mostly when touch-enabled LCD display is used. Since the images of input characters are overlaid with each other on the LCD display, it becomes difficult for the user to see what characters he / she is actually writing.

In the present study, we focus on the first problem, i.e. the segmentation problem in the overlaid handwriting recognition, and postpone the second problem, because the second problem can be easily solved by putting a short life time on each stroke echoed back.

This paper is organized with four sections. The next section describes the proposed method of overlaid handwriting recognition, and section three reports experimental evaluation. The final section is the conclusions.

## 2. Overlaid-Handwriting Recognition

The crucial difficulty in overlaid-handwriting recognition comes from the fact that boundaries between characters and boundaries between character components are not given explicitly to the recognition engines. From a technical point of view, the overlaid-handwriting recognition is a special case of continuous character recognition in which users write characters continuously. In contrast to the conventional continuous character recognition problem, location information of line segments does not help very much in the proposed framework.

The outline of the proposed system is shown in Fig. 2. The system basically consists of a feature extractor, sub-stroke models (HMMs), a hierarchical dictionary, a statistical language model, and a decoder that finds the most likely candidate sequence of characters.

### 2.1. Input Features

In the present study, pen position  $(x, y)$ , which is given by an input device, is used as the primary information. Let  $(\Delta x, \Delta y)$  be the difference between two consecutive pen-position samples and  $(r, \theta)$  be the feature vector, where  $r$  means the Euclidean distance between the two pen positions  $(\sqrt{\Delta x^2 + \Delta y^2})$  and  $\theta$  represents the direction of the feature vector. While the pen touches the tablet surface (pen-down), the feature vector  $(r, \theta)$  represents a velocity vector of the pen. When the pen leaves the tablet surface (pen-up),  $(r, \theta)$  represents a displacement vector between the strokes observed just before and after the pen-up, because the pen position is not sampled while it is in the air.

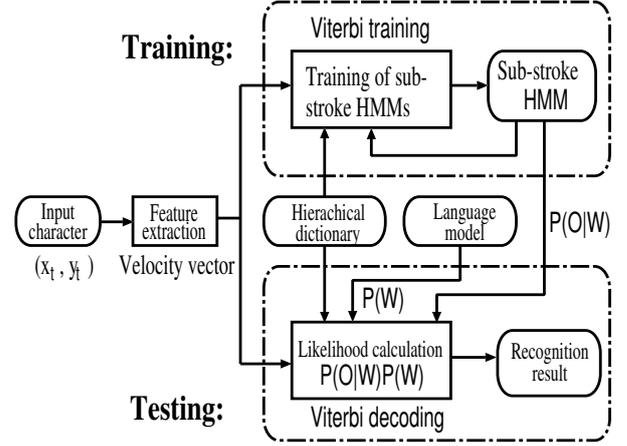


Figure 2. Overlaid-handwriting recognition system.

### 2.2. Recognition

As was noted previously, explicit character boundaries of input text are not given to the recognition engine; the recognition task is regarded as a continuous character recognition problem. Let  $\mathcal{W} = (W_1, W_2, \dots, W_N)$  be a candidate character sequence and  $\mathcal{O} = (\mathcal{o}_1, \mathcal{o}_2, \dots, \mathcal{o}_T)$  be the observation feature vector sequence. As the Bayes' theorem says, the recognition problem of finding the most likely character sequence  $\mathcal{W}^*$  when  $\mathcal{O}$  is given is defined as

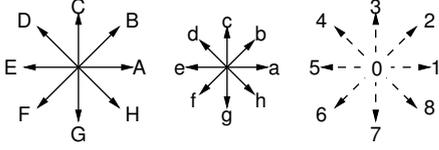
$$\begin{aligned} \mathcal{W}^* &= \arg \max_{\mathcal{W}} P(\mathcal{W}|\mathcal{O}) = \arg \max_{\mathcal{W}} \frac{P(\mathcal{O}|\mathcal{W})P(\mathcal{W})}{P(\mathcal{O})} \\ &= \arg \max_{\mathcal{W}} P(\mathcal{O}|\mathcal{W})P(\mathcal{W}). \end{aligned} \quad (1)$$

Here we calculate  $P(\mathcal{O}|\mathcal{W})$  and  $P(\mathcal{W})$  by using “sub-stroke HMMs” with a hierarchical dictionary, and a stochastic language model, respectively.

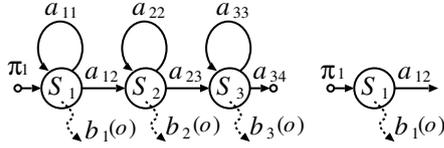
#### 2.2.1. Substroke HMMs

Unlike the English alphabet, there are more than ten thousand characters in the Japanese alphabet including Kanji and Kana. Because of this, it is not practical to model each character by an HMM so that the number of HMMs is identical to the number of distinct characters. Since Kanji characters have hierarchical structures and same components are shared between numbers of characters, a much smaller unit than a whole character can be chosen as the model unit. In the present study, substroke, that is the smallest unit of Kanji characters, has been chosen as the model unit [3].

Depending on the directions, lengths, and pen-up/down movements of substrokes, we define 25 substrokes of eight directions as shown in Fig. 3: eight long strokes (A–H), eight short strokes (a–h), eight pen-up movements (1–8) and



**Figure 3. Substroke categories: A–H (a–h) are long (short) substrokes with pen down, and 0–8 are the directions of pen up.**



**Figure 4. Substroke HMMs : pen down model (left), and pen up model (right).**

one pen-up-down movement (0). The substrokes are modeled by left-to-right continuous observation density HMMs as are shown in Fig. 4. A three-state HMM is employed for each pen-down substroke to model the changes in substroke velocity, while a one-state HMM without self-loop transition probability is used for each pen-up substroke to model the displacement vector.

Apart from the 25 substroke HMMs, an inter-character model is needed for the overlaid handwriting recognition. The inter-character model models the pen-up movement between the last pen-down stroke of the preceding character and the first pen-down stroke of the succeeding character. Even though the inter-character model would depend a lot on the characters concerned, the simplest model, i.e. the one-state HMM is employed, and it is named pen-up movement ‘9’.

Let  $\lambda^{(k)} = (A^{(k)}, B^{(k)}, \pi^{(k)})$  be the set of HMM parameters of substroke  $k$ , in which

$A^{(k)} = \{a_{ij}^{(k)}\}$  : The state-transition probability distributions from state  $S_i$  to  $S_j$ ,

$B^{(k)} = \{b_j^{(k)}(\mathbf{o})\}$  : The probability distributions of observation symbols  $\mathbf{o}$  at state  $S_j$ ,

$\pi^{(k)} = \{\pi_i^{(k)}\}$  : The initial state probability distributions.

The observation probability distribution is modeled by an  $M$ -mixture of Gaussian distributions:

$$b_i(\mathbf{o}) = \sum_{m=1}^M c_{im} \frac{\exp\left(-\frac{1}{2}(\mathbf{o} - \boldsymbol{\mu}_{im})^t \boldsymbol{\Sigma}_{im}^{-1} (\mathbf{o} - \boldsymbol{\mu}_{im})\right)}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}_{im}|}}, \quad (2)$$

where  $\boldsymbol{\mu}_{im}$  is the mean vector,  $\boldsymbol{\Sigma}_{im}$  is the covariance matrix, and  $c_{im}$  is the weighting coefficient (branching factor). Here, each Gaussian distribution is periodic with a  $2\pi$  cycle

with respect to the direction feature ( $\theta$ ).

In recognition, when an observation sequence  $\mathbf{O}$  is given, the probability  $P(\mathbf{O}|\mathcal{W})$  of a candidate character sequence  $\mathcal{W}$  can be calculated by an HMM which is comprised of the substroke HMMs according to the hierarchical dictionary described in the next section. Assuming that the HMM corresponding to  $\mathcal{W}$  is a concatenation of substroke HMMs and its sequence is denoted as  $\lambda_1, \lambda_2, \dots, \lambda_L$ , where  $\lambda_i$  represents the substroke number,  $P(\mathbf{O}|\mathcal{W})$  is given by

$$P(\mathbf{O}|\mathcal{W}) = \sum_{\text{all } q} P(\mathbf{O}, q|\mathcal{W}), \quad (3)$$

$$P(\mathbf{O}, q|\mathcal{W}) = \prod_{n=1}^L \left( \pi_{q_{T_{n-1}+1}}^{(\lambda_n)} \prod_{t=T_{n-1}+1}^{T_n} a_{q_{t-1}, q_t}^{(\lambda_n)} b_{q_t}^{(\lambda_n)}(\mathbf{o}_t) \right), \quad (4)$$

where  $q = q_1 q_2 \dots q_T$  is a state sequence that outputs  $\mathbf{O}$ ,  $T_0 = 0$  represents the start time, and  $T_L$  is equivalent to the time  $T$ .

## 2.2.2. Hierarchical Dictionary

The hierarchical dictionary provides the decoder with the information of how each character consists of the substrokes. The decoder processes an input pattern by referring to the character’s substroke sequence expanded from the hierarchical structure dictionary [3]. For example, the definition of the character “二” is ‘a 6 A’, which represents two pen-down strokes ‘a’ and ‘A’ connected with pen-up model ‘6’ in standard stroke order. Similarly, “子” is ‘A f 0 G d 4 A’ and “字” is ‘g 5 g 3 A f 6 A f 0 G d 4 A’, where “子” is a partial structure of “字” and they have a common substroke sequence.

According to the description in the dictionary, the decoder concatenates the substroke HMMs and generates a ‘character HMM’ corresponding to each candidate character.

## 2.2.3. Language Model

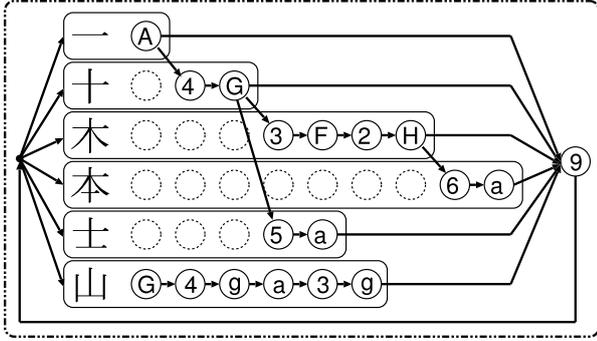
As a simple stochastic language model, a character bigram model is employed in the present study. Using the bigram model, the probability  $P(\mathcal{W})$  is given as

$$P(\mathcal{W}) = P(W_1) \prod_{i=1}^{N-1} P(W_{i+1}|W_i), \quad (5)$$

where the unigram probability  $P(W_1)$  is assumed to be independent from characters.

## 2.2.4. Network Search

In order to achieve fast and efficient search in recognition, network search strategy is employed. As is depicted in the



**Figure 5. An example of network structure for searching the best character sequence.**

example of Fig. 5, the network consists of parallel paths of state sequences, each of which stands for each character HMM, and the single state for the inter-character HMM. The final state of each character HMM is connected back to the initial state of the network by way of the inter-character pen-up model '9'. To save duplicated calculations, common states between the character HMMs are shared, and such states are marked with a blank circle without model name in the example.

Finding the optimal character sequence defined in (1) is implemented as a network search problem of finding the best path that maximizes the log likelihood defined as

$$L(W|O) = \log P(O|W) + L_W \log P(W) + N I_P, \quad (6)$$

where  $L_W$  is a weighting coefficient for the bigram language model, and  $I_P$  is the character insertion penalty. The One-pass beam search algorithm [7], which works time-frame synchronously, is used here so that temporal recognition results are obtained as the input goes on.

### 2.3. Training

The Viterbi training algorithm [1] is employed to train the substroke model  $\lambda$  with a set of training samples  $O$  so that  $P(O|\lambda)$  is maximized with respect to the model parameters: mean vectors  $\{\mu\}$ , covariance matrices  $\{\Sigma\}$ , branching factors  $\{c\}$ .

It should be noted that the substroke-level labeling is not available for the training samples, i.e. the training samples are not segmented into the sequence of substrokes. Furthermore, sample sequences of overlaid-handwritings are not segmented into the characters but only the character transcription is available.

For such insufficient training sample problems, the Viterbi-based embedded training (concatenated training) is employed to train the substroke HMMs. The inter-character pen-up model '9' is also trained in the same manner.

**Table 1. Experimental conditions**

Substroke HMMs	2-Gaussian mixtures (full covariance)
pen-down:	16 models for Kanji, 16 models for Hiragana
pen-up:	10 models
Search parameters	search width = 1000 $L_W = 4.0, I_P = -0.5$
Dictionary	1087 characters (1016 educational Kanji, and Hiragana)
Handwriting data	578 Japanese texts
Training:	30 people (set-A), 30 people (set-B)
Evaluation:	5 people (set-A), 30 people (set-B)

## 3. Experiments

The proposed method was evaluated through the recognition performance experiments.

### 3.1. Experimental Conditions

Overlaid-handwriting samples were collected from 35 people under the condition that they wrote given texts in arbitrary stroke orders using a stylus pen tablet (Wacom intuous i-400). The database ('set-A') consists of 578 Japanese texts whose character length is between 2 and 8.

Since the character recognition system used in this study does not take any measures against different stroke order problem<sup>1</sup>, recognition performance deteriorates when users write characters in different stroke order from the one defined in the hierarchical dictionary. Another database in which characters are written in fixed stroke order is needed to investigate how the different stroke order problem influences the recognition performance. For this purpose, an overlaid-handwriting database with fixed stroke order ('set-B') was artificially created from the isolated-character database where each character is written separately in the same stroke order as that of the dictionary.

Each database was split into two datasets: the training dataset for training the substroke HMMs, and the evaluation (testing) dataset for evaluating the recognition performance.

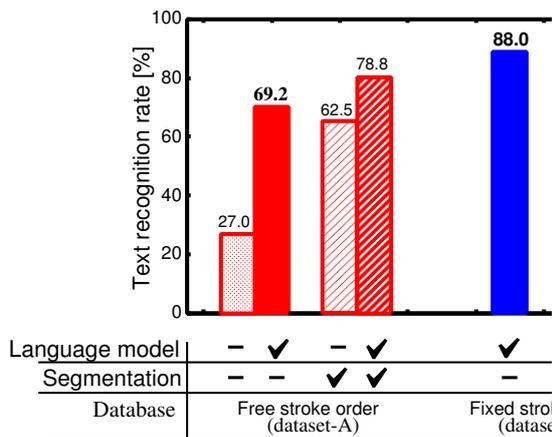
The character bigram language model was trained with the newspaper text corpus 'CD Mainichi Newspaper 1997 edition'. The test-set perplexity of the bigram was 134.1.

Details of the experimental conditions are shown in Table 1.

### 3.2. Experimental results

Recognition experiments were conducted using the two databases: set-A (free stroke order) and set-B (fixed stroke

<sup>1</sup>Although the authors have proposed a novel method [4] for the different stroke order problem, it has not been employed yet in this study.



**Figure 6. Recognition performance comparison**

order). Fig. 6 shows the experimental results where the vertical axis shows text-based correct recognition rate. The recognition rates are shown for different experimental setups: whether the bigram language model is used or not, and whether the character boundaries, i.e. the segmentation information, are given or not to the recognition engine. For example, 69.2% of evaluation texts of set-A were correctly recognized when the language model is used but no character-boundary information is used.

It can be said from the results that the bigram language model greatly helps to improve the recognition performance, especially when the character-boundary information is not available.

When language model is used and boundary information is not given, the proposed method gives the text recognition rate of 69.2% on set-A, while 88.0% on set-B, and character recognition accuracies of 74.9% and 91.1%, respectively. The main reason of such a big difference would be the different stroke order problem. In this sense, the recognition performance of the proposed method could be improved if various stroke orders were effectively added to the hierarchical dictionary.

Error analysis revealed that there are four types of errors depending on the reasons: (1) different stroke order, (2) similar characters, (3) language model mismatch, and (4) segmentation ambiguity. The type 1 and type 2 errors occur not only for the overlaid handwriting task but also for the usual isolated / continuous handwriting task, and such error would be reduced if the HMM based recognition engine was improved. The type 3 error could be reduced by employing better language models such as character trigram and word N-grams. The type 4 error would be partially reduced by the help of better language models, but some of the segmentation ambiguity cannot be resolved due to the fact that some part of a Kanji can be another Kanji, and

both hypotheses hold even if an N-gram language model is used.

## 4. Conclusions

The proposed approach based on the substroke HMMs and a statistical language model can be applied not only to the overlaid-handwriting recognition of characters that have hierarchical structures, but also to other on-line handwriting recognition applications where explicit segmentation information is not available. For example, it can be applied to “write anywhere” and “write in any directions” interfaces.

To tackle the segmentation ambiguity problem, the authors are going to investigate the use of auxiliary features such as pen pressure [5] and context-dependent models [9].

## References

- [1] HTKBook. <http://htk.eng.cam.ac.uk/docs/docs.shtml>.
- [2] Graffiti. <http://www.palm.com/products/input>.
- [3] M. Nakai, N. Akira, H. Shimodaira, and S. Sagayama. Substroke Approach to HMM-based On-line Kanji Handwriting Recognition. In *Proc. of Int. Conf. on Document Analysis and Recognition (ICDAR'01)*, pages 491–495, Sep 2001.
- [4] M. Nakai, H. Shimodaira, and S. Sagayama. Generation of Hierarchical Dictionary for Stroke-order Free Kanji Handwriting Recognition Based on Substroke HMM. In *Proc. of Int. Conf. on Document Analysis and Recognition (ICDAR2003)*, Aug 2003.
- [5] M. Nakai, T. Sudo, H. Shimodaira, and S. Sagayama. Pen Pressure Features for Writer-Independent On-Line Handwriting Recognition Based on Substroke HMM. In *Proc. of Int. Conf. on Pattern Recognition (ICPR2002), III*, pages 220–223, Aug 2002.
- [6] K. Nathan, H. Beigi, J. Subrahmonia, G. Clary, and H. Maruyama. Real-time On-line Unconstrained Handwriting Recognition Using Statistical Methods. *Proc. ICASSP95*, 4:2619–2623, 1995.
- [7] L. Rabiner and B. Juang. *Fundamental of Speech Recognition*. Prentice Hall, 1993.
- [8] G. Seni. TreadMill Ink – Enabling Continuous Pen Input on Small Devices. In *8th International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, pages 215–220, 2002.
- [9] J. Tokuno, N. Inami, S. Matsuda, M. Nakai, H. Shimodaira, and S. Sagayama. Context-Dependent Substroke Model for HMM-based On-line Handwriting Recognition. In *Proc. Int. Workshop on Frontiers in Handwriting Recognition (IWFHR-8)*, Aug 2002.
- [10] Transcriber. <http://www.microsoft.com/mobile/pocketpc/downloads/transcriber.asp>.