Overlapped handwriting input on mobile phones

Yanming Zou, Yingfei Liu, Ying Liu, Kongqiao Wang
Nokia Research Center, Beijing, BDA, 100176 P.R.China
yanming.zou, yingfei.liu, ying.y.liu, kongqiao.wang@nokia.com

***Abstract*—In this paper, we propose an overlapped handwriting input method on handheld devices, which allows users to write continuously without breaks on a single size-restricted writing area. 2 issues have been considered during the implementation of the overlapped input method: previous characters on the background may obstruct the clear viewing of current character and the messy overlapped handwriting is difficult to be segmented and recognized. In our method, a quick segmentation method based on an artificial neural network is used to tackle the first problem and a novel system is implemented to recognize the messy handwriting based on the output of an isolated character recognition engine and a language model. The recognition rate for Chinese characters is about 92.5% for a testing database containing GB2312 Chinese characters and other frequently used symbols. The positive feedbacks from testers have also confirmed the validity of the proposed method.**

***Index Terms*—overlapped handwriting recognition, artificial neural network, language model**

I. Introduction

Writing is always considered as a natural and efficient way for Chinese input on touch devices. Intensive researches have been held in this field since 1960s [1]. In 1990s, isolated character recognition has already been supported on some PCs and handheld devices. Nowadays, users can write a word [2] or even a whole sentence [3] continuously on a touch screen without breaks, just like writing on a piece of paper. With the help of natural language processing technologies [4], the total accuracy of continuous handwriting recognition is larger than 90%, which is similar to or even higher than the accuracy of common isolated character recognition.

However, the screen size restricts the usage of continuous handwriting input method on handheld devices, especially when users want to write with their fingers. For example, on a common 3.5 inches capacitive touch screen, it is difficult for users to write more than 3 simplified Chinese characters with fingers at a time, even if the whole screen is set as a writing area. Comparing to the whole sentence input experience on PC, the efficiency of the continuous handwriting input is lowered dramatically on size-restricted handheld devices.

One solution to support users writing characters continuously on such a small touch screen is to allow those characters being overlapped with one another. And then the resulted overlapped messy handwriting will be segmented and recognized automatically as characters. However, this will cause 2 main problems. The first is that since there is no spatial interval between characters, the segmentation and recognition process becomes more difficult for overlapped handwriting than common continuous handwriting. The second is that since the handwritten characters are all displayed at the same location, users can not see clearly what he/she is currently writing. Users may be confused and slow down the writing speed or even stop writing and wait for the screen to be cleared.

There have already been some researches on overlapped character segmentation and recognition. In [5], Shimodaira et al. introduced a substroke HMM based method for overlapped Japanese character recognition. It can recognize 1016 Japanese educational Kanji and 71 Hiragana characters, and the recognition rate is about 69.2% when different stroke order was permitted. In [6], Tonouchi et al. proposed a system to recognize overlapped handwritten Hiragana characters and convert those Hiragana characters to Kanji characters with special designed gestures. It shows that the efficiency of the proposed system is not lower than the efficiency of systems based on Multi-tap KKC method or multi-box handwriting input method. In [7], Bharth proposed an overlapped handwriting recognition system for English words in lowercase and word recognition accuracy is about 90% when there are 20K words in the lexicon. However the results mentioned above cannot satisfy the requirement of Chinese users. Currently, a product-level Chinese handwriting recognition engine needs to support at least 6716 characters, defined by the national standard GB2312, recognize at least 10 characters per second with a correct rate larger than 90% for unconstrained handwriting samples. This means some research works are still needed here.

To let users view his current handwriting clearly, one common solution is to let the strokes fade gradually during the writing process with a pre-defined fading speed. But the effect of this method is not good enough, since a fixed speed cannot deal with the diversity of different use cases. For example, when the writing character has a simple structure and a few short stokes, the next character may be overlapped on it before it totally fades out. While in other cases when the character is complex and has lots of long strokes, the initial part of the character may fade out before the entire character is finished. Thus an adaptable fading method which can detect the beginning of one character is needed for a good overlapped handwriting input method.

In this paper, we propose an input method to overcome the 2 problems mentioned above for simplified Chinese characters. Our method has been implemented on Nokia phones and experiments have showed that the recognition algorithm and

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.82

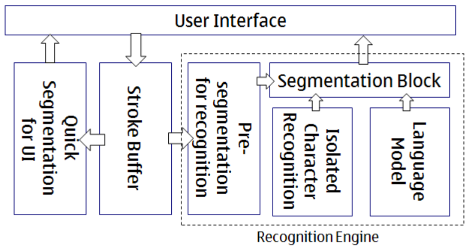

Fig. 1. Framework of Proposed Method

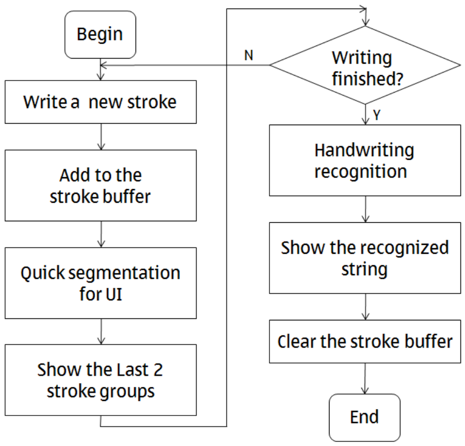

Fig. 2. Interaction Flow Chart

user interface of our method are good enough to support efficient Chinese input on handheld devices.

The rest of this paper is organized as follows. Our method is introduced in section 2. Section 3 gives out the evaluation results and section 4 concludes the paper.

II. Proposed Method

The framework of the proposed method is shown in (Figure1), which contains a well designed user interface, 1 stroke buffer for storing input strokes and quick segmentation results, 1 quick segmentation block to control the fading speed of handwritings and 1 handwriting recognition engine to give out the final text string corresponded to the input strokes. The quick segmentation method is also used in the recognition engine to speed up the whole recognition process.

The interaction flow of the input process is as follows (Figure2). When a stroke is finished, it will be stored in the stroke buffer and then sent to the quick segmentation block to check if it is the first stroke of a new character. If it is, a new quick segmented group will be generated and stored in the stroke buffer. Only the last 2 quick segmented groups will be shown on the screen with different grey levels, while other groups will be invisible for users. After the writing process is stopped, the recognition result will be given out by the recognition engine in a few seconds.

The details of each block will be given in the following subsections.

A. *Quick segmentation for UI*

The purpose of using a quick segmentation block for user interaction is to help user to view what he is currently writing clearly by removing unneeded strokes from the screen. The ideal result is that only the strokes of currently written character are shown on the screen. This means the method needs to segment the input stroke group and find out all the strokes of currently written character before a user starts to write the next stroke. Thus the running time left for the method for each input stroke is only the time slot between the 2 adjacent strokes, which is usually shorter than 0.1 second.

Our quick segmentation method is based on the detection of segmented strokes, which is defined as the first stroke of a character. Thus the input stroke group can be divided into small groups, which contains and only contains one segmented stroke and all the non-segmented strokes between it and the next segmented stroke. To speed up the segmentation even more, it is assumed that the detection result of one stroke will not be changed when new strokes are added to the whole group. Thus the detection for one stroke is only needed as the stroke has just been written.

Although all the strokes are overlapped altogether, it is still possible to make such a detection quickly based on only some spatio-temporal features of a stroke. One important fact is that, since most of the Chinese characters are written from left to right and from top to right, the probability for a stroke to be a segmented stroke is very high, if the stroke locates at the top-left side of its previous.

2 statistical classification methods were checked for this detection problem: support vector machine (SVM) and artificial neural network (ANN). Some experiments [8] show that SVM methods can give more accurate result than neural networks, but also require more computation resources. Since speed is more important than accuracy for our use cases, we finally selected a 3-layer neural network as our statistical classifier for quick segmented stroke detection, which can be expressed as.

$$\begin{aligned} O &= \sigma\Big(\sum_{\alpha=1}^{n} c_\alpha \sigma(\boldsymbol{w}_\alpha \cdot \boldsymbol{v} + b_\alpha)\Big) \\ \sigma(\mu) &= \frac{2}{1+exp(-\mu)} - 1 \end{aligned} \tag{1}$$

In our method, the neural network has an input layer composed of an input vector $\boldsymbol{v}$, and a middle layer with $n = 30$ units and 1 output unit. The target value for segmented stroke is set as -1, while that of non-segmented stroke as +1. The coefficients of the network are trained with a general back propagation training method.

The feature vector $\boldsymbol{v}$ used here consists of 20 features, which are extracted from the concerned stroke and the one written before and after it respectively. The features includes the boundary positions of the 3 stroke's bounding boxes, the positions of the starting point and geometric center of the concerned stroke, the position of the ending point and geometric center of the previous stroke. All those features

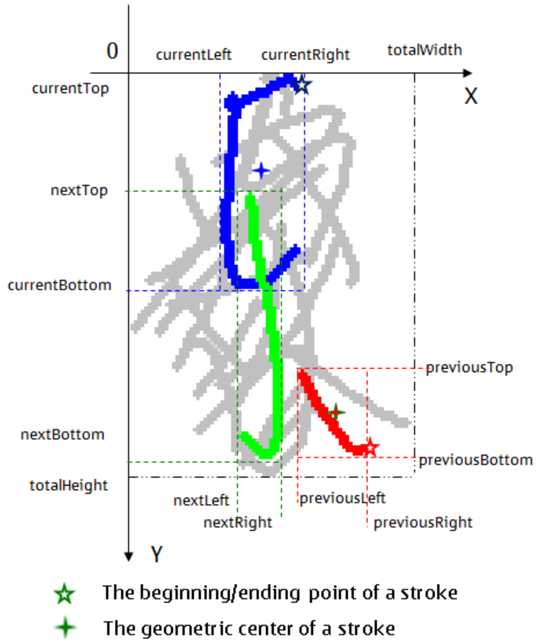

Fig. 3. Features for segmented stroke detection

are linearly normalized to a range from 1 to -1 based on the bounding box of all the strokes at hand. Since there is no stroke before the first stroke of the whole input group, all related features of the first stroke are set as -1. The same happens to some features of the last stroke. For details, please refer to the (Figure3) and our patent [9].

A threshold th is needed here to give the final decision. If the output of the neural network is smaller than th, the stroke is a segmented stroke, otherwise it is not.

Actually there are two kinds of errors related to the detection,

- A non-segmented stroke is classified as a segmented stroke;
- A segmented stroke is classified as non-segmented stroke.

The problems caused by the 2 kinds of errors are different during the writing process. For the first kind of errors, the problem is that part of current written character becomes totally invisible. And the problem caused by the second kind of errors is that different characters are overlapped together and then it is difficult for users to view what he is writing clearly. It is obvious that the problems caused by the first kind of errors are much more serious than those caused by the second. Thus a proper threshold should prevent the happening of the first kind of errors while keep the possibility of the second kind of errors at an acceptable level. In the real system, to reduce the effect of the first kind of errors even more, we keep the last 2 quick segmented groups on the screen with 2 different grey levels respectively.

B. Handwriting recognition

Similar to the common continuous handwriting recognition [10], overlapped handwriting recognition can also be described as an optimization problem to extract the text string with maximum a posteriori probability from a sequence of written strokes, which can be expressed as:

$$C_{max} = arg \max_{C_i \in \mathbf{C}} \max_{G_j \in \mathbf{G}} P(C_i|G_j) \tag{2}$$

where C_i represents one possible candidate in the text string collection $\mathbf{C}$ and $G_j = \{g_j^0, \ldots, g_j^k \ldots, g_j^{N-1}\}$ is a segmentation scheme of the input stroke sequence. Here we use g_j^k to represent a segmented stroke group in the segmentation scheme G_j.

One way to solve the problem above is to calculate the probability of each text string for each segmentation scheme directly. The advantage is that the best segmentation scheme and the best text string can be got simultaneously in one step. However, its computation consumption is too high to be supported even on desktop computers, let alone handheld devices. To make the algorithm workable on handheld devices, we lower the computation consumption by reducing the number of possible schemes to be evaluated and simplifying the evaluation process of each segmentation scheme. The result system is shown in (Figure1), which has an efficient pre-segmentation block and a common segmentation block supported by isolated character recognition engine and a bigram language model. The two blocked will be introduced in the following paragraphs.

1) Presegmentation: The purpose of pre-segmentation block used here is to dramatically reduce the searching space of the segmentation block. This is done by first dividing the whole input stroke sequence into some small stroke groups and then asking the segmentation block to only evaluate the segmentation schemes composed by those pre-segmented groups.

The task here is similar to the quick segmentation task for user interactions. So we used the same neural network here with a different strategy to choose a proper threshold for the classifier. To reach a high accuracy of the whole system, it is required to keep the correct scheme in the searching space, which means the output of pre-segmentation block should be a whole character or a part of character. Thus we should choose a threshold which can prevent the second type of errors while reducing the first type of errors as much as possible

2) Segmentation: Although the presegmentation block reduces the number of segmentation schemes dramatically, it is still unaffordable to going through the whole text string collection for all the combinations of pre-segmented groups. To further reduce the computation consumption, we propose a 2-step method to solve this problem. The first step is to find the best method to segment the strokes into written characters and then in the second step to find the text string with the maximum a posteriori probability corresponded to the scheme.

The proposed method can be expressed as

$$G_{max} = arg \max_{G_i \in \mathbf{G}} \Pi P(g_i^k) \cdot P(g_i^k | g_i^{k-1})) \quad (3)$$

$$C_{max} = arg \max_{C_i \in \mathbf{C}} \Pi P(c_i^k | g_{max}^k) P(c_i^k | c_i^{k-1}) \quad (4)$$

Here, $P(g_i^k)$ represents the likelihood of a stroke group g_i^k to be a valid character. Its value can be calculated by

$$P(g_i^k) = \alpha(g_i^k) \cdot \max_j P(c_i^{k,j} | g_i^k) \quad (5)$$

in which $P(c_i^{k,j} | g_i^k)$ represents the probability of the stroke group g_i^k to be the character $c_i^{k,j}$, which is given out by the isolated character recognition engine, while the parameter $\alpha(g_i^k)$ is decided by the geometrical relations between stroke group g_i^k's and other strokes written before and after it. $P(g_i^k | g_i^{k-1})$ is the probability of stroke group g_i^{k-1} to be followed by g_i^k in an usual handwriting input experience. It is defined as

$$P(g_i^k | g_i^{k-1}) = P(c_m^k | c_n^{k-1}) \quad (6)$$

where(m, n) is decided by

$$\begin{aligned} (m, n) &= arg \max_{(j,l)} P(c_i^{k,j}, c_i^{k-1,l} | g_i^k, g_i^{k-1}) \\ &= arg \max_{(j,l)} P(c_i^{k,j} | c_i^{k-1,l}) \cdot \\ &\quad P(c_i^{k,j} | g_i^k) \cdot P(c_i^{k-1,l} | g_i^{k-1}) \end{aligned} \quad (7)$$

Just as above, $P(c_i^{k,j} | g_i^k)$ and $P(c_i^{k-1,l} | g_i^{k-1})$ are provided by the isolated character recognition engine, while $P(c_i^{k,j} | c_i^{k-1,l})$ is given by the bigram language model.

This method dramatically reduces the computation complexity of the original problem. Details of algorithm implementation and computation complexity analysis can be found in reference [10]. (Figure 4) gives out an example of the recognition processing of an input stroke sequence.

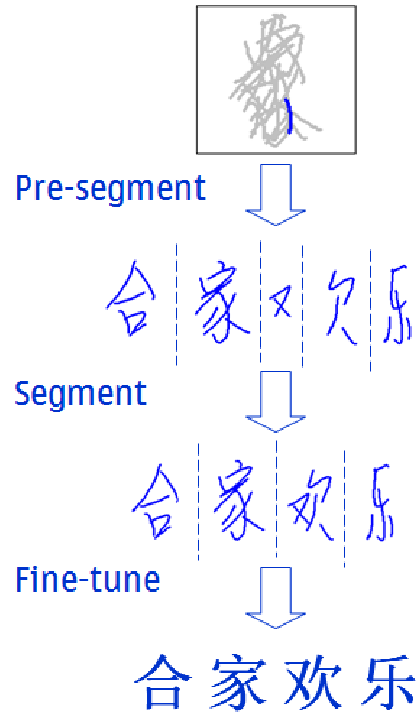

Fig. 4. Recognition Process

III. Experiment and Evaluation

To build and test the performance of our method, we have collected an overlapped handwriting database, which contains about 1500 sentences from 50 people. The average number of characters contained in one sentence is about 15. Most of the characters in the database are simplified Chinese characters, but some frequently used symbols, like punctuations, English characters and digits, are also included.

The whole database is divided into 3 parts for our experiment. The first part which contains about 500 sentences were used for training the quick segmentation algorithm, another 500 sentences were taken as reference to fine-tune the parameters of the whole algorithm, and the remaining samples were used for testing.

Most of the simplified Chinese characters contain multiple strokes. If only one sample is extracted for each stroke, there will be much more samples for non-segmented strokes than those for the segmented strokes. This is OK for testing sample set. But for training set, more segmented samples are needed. To achieve this purpose, the combinations of the one character and all the other characters in front of it in a sentence are utilized. For example, in a sentence consisting of ABC 3 characters, 2 segmented samples are extracted for character C, which are based on the character combinations of AC and BC respectively. The numbers of training and testing samples are given in table I. The error rates of the trained neural network for different threshold are given in (Figure5).

According to the discussion in previous paragraphs, we set the quick segmentation threshold for UI interaction as -0.5, which grantees that 95% characters will not be divided into 2 or more parts in the writing process. The threshold for the pre-segmentation part for handwriting recognition is set as 0.5, which means 98% of the pre-segmented stroke groups satisfy the requirement of the segmentation block.

To evaluate the total performance of the whole system, a common isolated character recognition engine is licensed from an outside company. Its recognition range includes 6763 simplified Chinese characters, punctuations, English characters and digits. The language model is trained with the text from prevalent Chinese newspapers and covers the same character set. The test result given in table II shows that the recognition rate of our overlapped handwriting recognition is similar to the isolate character recognition engine. One drawback of the algorithm is about the symbols whose size is relatively smaller than normal character, like symbol comma ',' and full stop'.'. It is very difficult to segment it from the character overlapped on it.

A user study was conducted to check user acceptance of the

TABLE I
TRAINING AND TESTING SETS

	Training Set	Testing Set
Segmented samples	75181	12699
Non-segmented samples	22221	46221

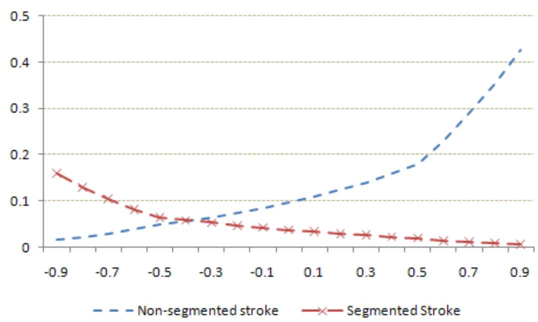

Fig. 5. Error rate of the quick segmentation

overlapped Chinese HWR solution. Five experienced users and 5 novice users of the discrete Chinese HWR methods with mobile devices took part in the study. Participants needed to complete two tasks: the first task was to enter Chinese characters freely with the overlapped Chinese HWR UI, which was to check if participants can discover the feature; and before the second task, the study moderator would introduce the continuous writing feature to participants and ask them to enter a few pre-defined Chinese text messages. After both tasks were complete, we would ask the participants to evaluate the method with a 5-point Likert questionnaire to indicate their satisfaction level with the feature, and 5 is the most desired level. The result shows that only 1 user can discover the overlapped input feature by himself. But after we reminded the rest participants of the continuous writing feature and they tried it, they gave respectively 4.3 (SD = 0.67) and 4.2 (SD = 0.63) on its usefulness and ease of use .

We also conducted a comparative user study to understand user performance (text entry rate and error rate) with OHI and other four Chinese HWR methods on mobile devices including 2 common continuous and 2 isolated HWR methods. Here, common continuous HWR methods mean that users can write several characters on the screen continuously, just like the method introduced in [10]. The results showed that our method is significantly faster than three of the four methods. However, users made more errors with the 3 continuous HWR methods than with the two isolated HWR methods. Within each category of methods, there is no significant differences found on error rate. The study results also indicated that there is no significant difference between finger HWR and HWR with stylus.

IV. CONCLUSION

The overlapped handwriting input method proposed in this paper can support users to write continuously in a small writing area with characters overlapped with each other. During the writing process, our method can detect the beginning of each character, and then let the handwriting of previous characters fade out. Thus users can view his/her current writing clearly. When the writing process is finished, accurate recognition result will be given out very quickly. The whole system has already been implemented on a Nokia phone. Evaluation shows that the novel input method can provide more efficient input experience for users than the common isolated character input method. In future, the support for small-size symbols and function reminding by the application itself needs to be enhanced.

TABLE II
RECOGNITION RATE

Recognition rate	Overlapped Characters	Isolated Characters
Chinese characters	92.5%	93.0%
Small symbols (.,.:)	31.1%	42.87%
Other symbols	73.5%	73.7%

ACKNOWLEDGMENT

The authors owe special thanks to the colleagues in Nokia Research Centre Beijing, who have facilitated the research with generous help and advice. We also express our appreciations to the Centre for Intelligent Image and Document Information Processing in Tsinghua University for their technical discussions.

REFERENCES

[1] R. Dai, C. Liu, B. Xiao, "Chinese Character Recognition: History, Status and Prospects," FRONTIERS OF COMPUTER SCIENCE IN CHINA, vol. 1(2), pp. 126-136, 2007

[2] K. Ding, L. Jin, X. Gao, "A New Method for Rotation Free Method for Online Unconstrained Handwritten Chinese Word Recognition: A Holistic Approach," ICDAR, pp.1131-1135, 2009 10th International Conference on Document Analysis and Recognition, 2009

[3] H. Yan, L. Jin, C. Viard-Gaudin, H. Mouchre, "SCUT-COUCH Textline_NU: An Unconstrained Online Handwritten Chinese Text Lines Dataset," ICFHR, pp.581-586, 2010 12th International Conference on Frontiers in Handwriting Recognition, 2010

[4] C. Manning, H. Schtze: Foundations of Statistical Natural Language Processing, MIT Press, 1999

[5] H. Shimodaira, T. Sudo, M. Nakai, S. Sagayama, "On-line Overlaid-Handwriting Recognition Based on Substroke HMMs," ICDAR, vol. 2, pp.1043-1047, Seventh International Conference on Document Analysis and Recognition (ICDAR'03) - Volume 2, 2003

[6] Y. Tonouchi, A. Kawamura, "Text Input System Using Online Overlapped Handwriting Recognition for Mobile Devices," ICDAR, vol. 2, pp.754-758, Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2, 2007

[7] A. Bharth and S. Madhvanath, FreePad: A Novel Handwriting-base Text Input for Pen and Touch Interfaces, IUI 08 Proceedings of the 13th international conference on Intelligent user interfaces, pp. 297-300, 2008

[8] B. Zhu and M. Nakagawa: Segmentation of On-line Handwritten Japanese Characters of Arbitrary Line Direction Using SVM, Proc. 2nd Korea-Japan Joint Workshop on Pattern Recognition (KJPR2007, PRMU2007-115), Vol.107, No.281, pp.135-140, (2007.10)

[9] Y. Zou, Y. Chen, C. Liu and X. Wan, METHOD AND APPARATUS FOR SEGMENTING STROKES OF OVERLAPPED HANDWRITING INTO ONE OR MORE GROUPS, patent application: No. PCT/CN2010/076285

[10] Y. Zou, K. Yu and K. Wang, Continuous Chinese handwriting recognition with language model, Frontiers in Handwriting Recognition, 11th International Conference on, pp. 344-348, 2008