UDC 001.81:681.327

Interactive Character Recognition Technology for Pen-based Computers

●Kazushi Ishigaki ●Hiroshi Tanaka ●Naomi Iwayama

(Manuscript received June 11, 1999)

This paper describes Fujitsu's latest interactive (i.e., online) handwriting character recognition (OLCR) technology. To compensate for stroke order and stroke connection variations of handwritten Japanese characters, we have developed a hybrid character recognizer which integrates an online and an offline (OCR) recognition module. In an experiment, our hybrid recognizer achieved an 86.8% recognition accuracy while the online and offline modules scored 84.3% and 72.4%, respectively. The hybrid recognizer can recognize more than 4400 Japanese characters using a 400 to 800 KB dictionary. The recognition speed is about 30 ms per character with a P2/266 MHz CPU.
For context processing, we have developed a character-class Bi-gram based context processing module. In an experiment, our context processing module improves the recognition accuracy from 82.7% to 90.5% for non-kanji characters and from 90.6% to 93.8% for kanji characters with only a 40 KB dictionary.
Also, to improve recognition accuracy for a specific user, we have developed an adaptive context processing (ACP) technology which lets the system automatically learn user-input strings and applies the information to the context processing. In an experiment, we observed that the recognition accuracy was improved from 86.1% to 95.4% when the ACP was applied.

1. Introduction

The recent, remarkable progress of information technology and the severe business environment have been forcing many companies to use mobile computers and customer-direct operation systems (or self-service terminals) to save time and personnel expenses. Mobile computers are often required to be usable by one hand, therefore pen-input is often used instead of a keyboard. For pen-based computers, handwriting character recognition is one of the major methods used for character input. There are other methods for pen input,[1] for example, Software Keyboard, T-Cube,[2] and Quikwriting,[3] however, character recognition is the most natural method. Also, we believe it is the most effective method for untrained users, provided that the recognition accuracy is satisfactory. Finger touch operation is widely used on self-service terminals, and handwriting character recognition is expected to expand the application area of these terminals.

In this paper, we describe Fujitsu's latest interactive (online) handwriting character recognition (OLCR) technology for pen-based computers. We first explain some distinctive features of Japanese OLCR that are not found in alphabetic OLCR. We then describe our high-speed and high-accuracy hybrid character recognizer, in which we integrate an online and an offline recognition module. We also describe our highly compact and effective context processing module, which is based on the concept of the character-class Bi-gram. Then, we describe our prominent adaptive context processing technology, which enables a system to gradually improve its OLCR accuracy.

富士通は新しいオンライン文字認識ソフトを開発した。
K K K h K h h k k k k k K K K K k k k h K K h h s
(Fujitsu developed a new online handwritten character recognition software).

川崎市中原区上小田中４－１－１
K K K K K K K K K K n s n s n
(Fujitsu's address)

K: Kanji
h: Hiragana
k: Katakana
n: Number
s: Symbol

Figure 1
Example of Japanese text.

2. Features of Japanese OLCR and its technical requirements

The technical requirements for Japanese OLCR are somewhat different from those of alphabetic OLCR. The distinctive features of Japanese OLCR are as follows:

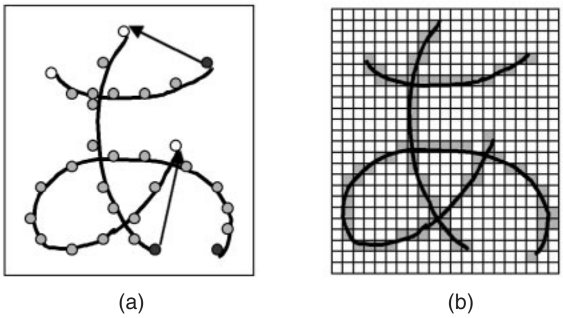
1) The number of recognition categories is very large. Typically there are more than 3000 (JIS-1st level) or 6000 (JIS-2nd level) characters.
2) Japanese text consists of various types of characters, for example, kanji, hiragana, katakana, alphabetic letters, and Arabic numbers. These types are combined in the same sentence (**Figure 1**).
3) Many of the kanji characters, which occupy the major percentage of recognition categories, have a rather large number of pen strokes (some have as many as 30). The writing orders of the strokes often differ between users. Also, stroke concatenation within a character is common and varies even with the same user.
4) Hiragana and katakana characters consist of only a few strokes, and their writing orders are more stable. However, there are similar looking characters within each of these two character sets. Also, some hiragana and katakana look the same, and some katakana and kanji look the same.
5) Japanese words are not separated by spaces.
6) Most Japanese are comfortable writing characters separately in handwriting boxes; therefore, in most cases, there is no segmentation problem with Japanese OLCR.

For alphabetic OLCR, word-based recognition, in which cursively written words are recognized by matching the input to a word dictionary, is popular. In alphabetic OLCR, the number of recognition elements (characters) is small, but the deformation of characters can be very pronounced and segmentation of characters is essential. To compensate for the low recognition accuracy of each character, alphabetic OLCR inevitably needs to use word information. On the other hand, most practical Japanese OLCR systems are character-based systems which recognize characters separately input into handwriting boxes. Some Japanese OLCR systems can recognize a sequence of characters without handwriting boxes, but their recognition accuracy tends to be poor. As a result most Japanese do not mind using handwriting boxes, so Japanese systems generally use character-based recognition. We want to emphasize that the use of character-based recognition does not indicate a technical backwardness in Japanese OLCR; rather, its use is dictated by the special nature of the Japanese system of writing.

Based on the characteristics of Japanese handwritten characters described above, we can summarize the major technical problems of Japanese OLCR as follows:

1) How to recognize a large number of characters while compensating for variations in writing order, stroke concatenation, and shape deformation, especially for kanji characters.
2) How to discriminate between similar characters, especially non-kanji characters.

To solve the first problem, we have developed a high-performance hybrid handwriting character recognizer which can recognize more than 4400 characters regardless of the writing order, stroke concatenation, and deformation. To solve the second problem, we have developed a high-performance context processing module which can



(a) Stroke sequence : online type.
(b) Bitmap image : offline (OCR) type.

Figure 2
Character recognition objects.

Table 1
Characteristics of recognition algorithms.

	Strong points	Weak points
Online type	Deformed shape Compact dictionary size	Writing order, Direction, Overwriting Noise.
Offline type	Writing order, Direction, Overwriting, Stroke counts Noise.	Deformed shape Similar characters

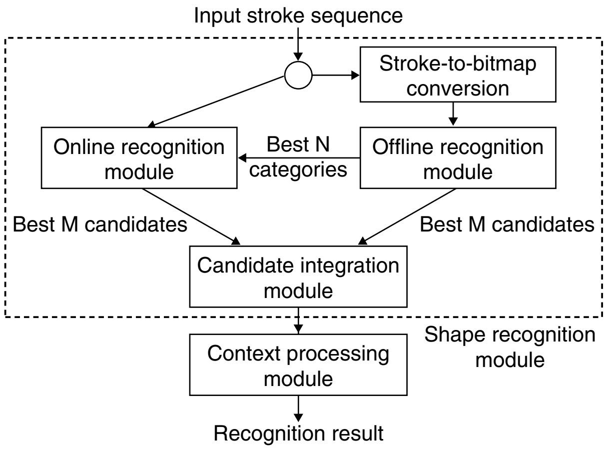


Figure 3
Hybrid character recognition system.

discriminate between similar characters using a very small dictionary.

3. Fujitsu's OLCR technology

3.1 Online recognizer and offline recognizer

The recognition object of OLCR is a time sequence of 2-dimensional points that mark the motion of the pen tip (**Figure 2 (a)**). The input order of the strokes is the writing order of the character. Since the writing order provides useful and distinctive information, most previous OLCR algorithms (we call them online algorithms) use this information to recognize characters. However, since the writing order is not so stable and has more variation than we could previously store in the recognition dictionary, these algorithms provided limited recognition accuracy.

Another character recognition technology called OCR (Optical Character Recognition) recognizes characters by their shapes (**Figure 2 (b)**). Since this type of recognition algorithm does not use the information about the writing order or stroke count, it is not affected by either of these variations unless they drastically deform the character image. Since the bitmap image of a character can easily be obtained from the OLCR input, OCR-type algorithms (we call them offline algorithms) can also be applied to OLCR. However, because they lack the stroke information, their recognition accuracy is usually inferior to that of online-type algorithms and they are especially weak for non-kanji characters unless a large number of matching templates are provided.

Table 1 summarizes the characteristics of these two types of recognition algorithms. The table shows that the two types have complementary characteristics. This table suggest the idea of combining the two types of recognition algorithms. Adopting this approach, we have developed a hybrid character recognizer[4] in which we integrated an online and an offline recognition module to compensate for the individual weaknesses of each type.

Figure 3 shows the block diagram of our hybrid handwriting character recognition system. It consists of an online recognition module, offline recognition module, candidate integration module, and context processing module.

3.2 Hybrid character recognizer

Online recognition module

The online recognition module uses a high-performance online recognizer developed at Prof. Nakagawa's Laboratory at the Tokyo University of Agriculture and Technology (TUAT). The recognition algorithm of this online recognizer is called "LTM" (Linear-Time elastic Matching).[5),6)] LTM recognizes characters by dynamically determining the correspondence of feature points of the input pattern with those of dictionary templates.

LTM successively determines the correspondence of feature points to maximize the local evaluation value $E(i, j)$ of the i-th input feature point $p_i = (X_i, Y_i)$ and the j-th dictionary feature point $q_j = (U_j, V_j)$. $E(i, j)$ as follows (**Figure 4**).

$$E(i,j) = \text{(Positional evaluation value)} \times \text{(Directional evaluation value)} = E_{\text{pos}}(i,j) \times E_{\text{dir}}(i,j),$$

where

$$E_{\text{pos}}(i,j) = \begin{cases} 2N - D_{\text{cb}}(i,j) \mid E_{\text{dir}}(i,j) \geq 0 \\ D_{\text{cb}}(i,j) \mid E_{\text{dir}}(i,j) < 0 \end{cases}$$

$$E_{\text{dir}}(i,j) = \cos\theta_{ij} = \frac{\overline{p_{i-1}p_i} \cdot \overline{q_{j-1}q_j}}{|\overline{p_{i-1}p_i}| \cdot |\overline{q_{j-1}q_j}|}$$

$$D_{\text{cb}}(i,j) = |X_i - U_j| + |Y_i - V_j|$$

N : Normalization size

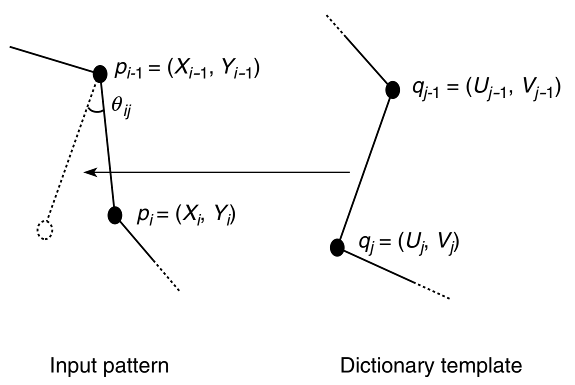


Figure 4
Evaluation value of correspondence.

LTM performs restricted-backtracking when the difference in length at the corresponding part is more than some threshold value. After determining the correspondence of all feature points, LTM calculates the final pattern-similarity value (see M. Nakagawa & K. Akiyama[5),6)] for more details).

LTM is a kind of elastic matching algorithm which is robust in terms of stroke concatenation. LTM is also strong regarding the shape deformation that occurs when characters are written very roughly. The dictionary of LTM is very compact since it is constructed from sub-patterns. LTM works like DP matching, but much faster. The original LTM recognizer was developed and improved by Dr. K. Akiyama by referring to Fujitsu's previous OLCR algorithm.[7)]

Offline recognition module

The offline recognition module uses a simplified and compressed version of Fujitsu's current OCR algorithm.[8)] We first generate a 64×64 bitmap image from the sequence of strokes by interpolating 1-width lines between the input sequence of points. Then, the input image is scanned in four directions and the bitmap densities within 6×12 (= 72) areas are calculated (**Figure 5**). This provides a 288-dimensional (= 72×4) feature vector of the input image.

By using the technique of canonical discrimi-

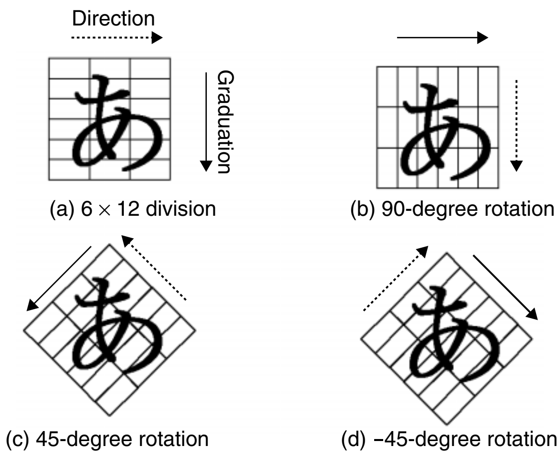


Figure 5
Feature extraction of offline recognition module.

nant analysis,[9] we then produce an N-dimensional (N< 288) compressed feature vector from the original 288-dimensional feature vector. The dictionary templates are a set of N-dimensional feature vectors similarly generated from an online handwritten character database. At first, to improve the recognition speed, a rough matching process is performed using the first M-dimensional part of each template (M< N: M is currently 16). Then, the detailed matching process using the full N dimensions is performed over the best L templates of the rough matching process (L is currently 800).

Table 2 shows the recognition performance of our offline recognition module for different numbers of compressing dimensions N. Dictionary templates were generated using the handwriting data of 75 people from the database "**nakayoshi**"[10] (HANDS-nakayosi-t-98-09). The recognition categories were 4430 characters (371 non-kanji, 2965 kanji-JIS1, and 1094 kanji-JIS2). The total number of learned characters was 478 039, with between 75 and 512 (maximum) characters for each category. The number of generated dictionary templates was 5386: approximately three templates for each non-kanji character and one template for each kanji character. The tested data are the Nos. 51, 55, and 66 of "**kuchibue**"[10] (HANDS-kuchibue-d-96-04). We excluded 11 punctuation characters, since they are usually too small for size-normalization to work correctly.

Table 2
Recognition results of offline module.

Dimensions	1st	2nd	5th	10th	100th	Dict.
32	61.4%	76.1%	87.1%	92.1%	98.8%	197 KB
48	67.8%	81.1%	90.5%	94.0%	99.1%	290 KB
64	70.4%	83.3%	91.6%	94.7%	99.1%	383 KB
96	72.4%	85.0%	92.5%	95.3%	99.1%	570 KB

Test data : HANDS_Kuchibue_d_97-06 No.51, 55, 66
Total : 33 888 characters (excluding 11 punctuation symbols)
Dictionary: 4430 char.: 2965 JIS1 kanji, 1094 JIS2 kanji, 371 non-kanji

Candidate integration module

The candidate integration module integrates the recognition candidates from the online and offline recognition modules. Candidate integration is performed after normalizing the offline recognition score so that it is comparable to the online score. For a recognition character category C_i, let $S_{\text{online}}(C_i)$ be the score of the online recognition module and $S_{\text{offline}}(C_i)$ be the score of the offline recognition module. Then, the normalized offline score $S_{\text{offline-normalized}}(C_i)$ and the final hybrid recognition score $S_{\text{hybrid}}(C_i)$ are obtained by:

$$S_{\text{offline-normalized}}(C_i) = (S_{\text{offline}}(C_i) - S_{\text{offline-mean}}) \times \mathrm{W} + \mathrm{H},$$
$$S_{\text{hybrid}}(C_i) = \mathrm{Max}\ (S_{\text{online}}(C_i),\ S_{\text{offline-normalized}}(C_i)),$$

where W and H are constants and $S_{\text{offline-mean}}$ is the mean score of the best 5th candidates of the offline recognition (see H. Tanaka[4] for more details).

To improve the recognition speed, we use the best K categories of the offline recognition module as a rough matching process of the online recognition module (K is currently 100).

Table 3 shows the recognition results of our hybrid OLCR (without context processing). By integrating the candidates of the two modules, the recognition accuracy is improved for both the 1st and N-th cumulative recognition rates. Our system recognizes over 4400 Japanese characters with a practical small dictionary (400 to 800 KB,

Table 3
Recognition results of hybrid recognizer.

Dimensions	1st	2nd	5th	10th	Dict.	Speed
32	86.0%	93.5%	97.0%	97.8%	425 KB	27 ms
48	86.3%	93.9%	97.4%	98.2%	518 KB	27 ms
64	86.6%	94.1%	97.6%	98.3%	611 KB	29 ms
96	86.8%	94.3%	97.7%	98.4%	798 KB	32 ms
LTM only	84.3%	91.6%	96.0%	97.2%	228 KB	

Test data : HANDS_Kuchibue_d_97-06 No.51, 55, 66, total 35 886 char.
(Kanji 5643, non-kanji 6319, total 11 962 for each)
Dictionary : 4441 char.: 2965 JIS-1 kanji, 1094 JIS-2 kanji, 382 non-kanji
Experiment environment: OS: Windows 95, CPU: P2/266 MHz, Memory: 96 MB

depending on the number of compressed dimensions). The recognition speed is approximately 30 ms per character using a P2/266 MHz CPU. This system is fast enough to run on a much slower CPU in a small hand-held terminal or a middle-class PDA.

3.3 Context processing module

The context processing (CP) module discriminates between similar characters after shape recognition using the context information. The word-based CP method is often adopted for OCR and alphabetic OLCR. However, for Japanese OLCR, word-based CP has not been considered effective because:

1) Japanese text is not usually separated by spaces. To separate a Japanese string into valid words may require morpheme analysis, which consumes lots of memory.
2) The total number of words required to cover general Japanese text would be very large. The memory requirement of the word dictionary would be too big for small pen-based computers.
3) To feed back the recognition results immediately, the CP module needs to be applicable even to very short strings, sometimes shorter than a single word or phrase.

To cope with these problems, a character (not word) Bi-gram based (or sometimes tri-gram) methodology[11] has been popular for context processing in Japanese OLCR. Character Bi-gram based CP corrects recognition results by using the probability of the transition (or appearance) of a continuous pair of characters. It can be applied to a very short string and runs fast. However, we think a conventional character Bi-gram based CP still has following disadvantages:

1) The effectiveness of the CP depends on the quantity of the learning corpus (database of text to be learned). Learning a large corpus will increase the dictionary size, but the performance tends to saturate. A typical dictionary often exceeds 1 MB.
2) The effectiveness of the CP is also sensitive to the type of learning corpus that is used. For example, a Bi-gram dictionary generated by a newspaper corpus is somewhat unsuitable for address inputs.

Table 4
Performance of context processing.

	With CP (1st)	Without CP (1st)	10th
Kanji	93.8%	90.6%	98.4%
Non-kanji	90.5%	82.7%	97.9%
Total	91.9%	86.3%	98.0%

Test data: Text part of HANDS_Kuchibue_d_97-06
Nos.51, 55, 66
(Kanji 3937, non-kanji 6217, total 10 154 for each)

Character-class Bi-gram based context processing

To compensate for the above problems, we have developed a character-class Bi-gram based CP module. Instead of using the probability of transition of characters, our CP module uses the probability of transition of character classes. A character class is a set of characters (most of them based on kanji, hiragana, katakana, etc.), but we have divided them into several sub-classes to improve correction performance. We are currently using about 100 character classes.

Our character-class Bi-gram based CP module corrects recognition results by using the probability of transition of character classes and the probability of appearance of the character.

Let the input sequence of a handwritten pattern be $X = X_1X_2 \cdots X_n$ and the corresponding character sequence be $C = C_1C_2 \cdots C_n$. We then define the evaluation value $L(C|X)$ as follows:

$$L(C|X) = \Sigma \mathrm{Sim}(X_i,C_i) + \omega\,(\Sigma \log P(C_i|T_i) + \Sigma \log P(T_i|T_{i-1})),$$

where $\mathrm{Sim}(X_i,C_i)$ is the recognition score of character C_i corresponding to input pattern X_i and T_i is the character class to which character C_i belongs. $P(C_i|T_i)$ is the probability of appearance of character C_i in character-class T_i. $P(T_i|T_{i-1})$ is the probability of transition of character classes from $T_{i\text{-}1}$ to T_i. And ω is a constant.

Our CP module calculates the evaluation values $L(C\,|\,X)$ for all possible character strings C which are generated by the sequence of recognition candidates of the shape recognition module. Finally, our CP module outputs string C that has the highest evaluation value.

By using this character-class Bi-gram based method, our CP module has been made very compact. Actually, our CP module uses only a 40 KB dictionary.

Table 4 shows some experimental results for our CP module. The test data is the text parts of the Nos. 51, 55, and 66 of "**kuchibue**" (the database consists of a text part and a random character part). The table shows that our CP module is especially effective for non-kanji characters. Since the frequency of non-kanji characters is rather high in Japanese text and the recognition accuracy of shape recognition modules for non-kanji characters is usually inferior when a deformation-free algorithm is used, effective context processing for non-kanji characters is in great demand.

4. Adaptive character recognition

In adaptive character recognition, the OLCR system learns to more accurately recognize the handwriting of a specific user over repeated inputs. There are two types of adaptation for OLCR; one is the adaptation of shape recognition and the other is the adaptation of context processing.

The idea of adaptation of shape recognition has been popular,[12),13)] and there are many OLCR systems which have a function for character shape learning. However, since character shape learning sometimes causes side effects and often requires a large memory for each learned character, there are only a few practical OLCR systems that permit automatic shape learning. In other words, they need the user to perform explicit learning (or registering) operations.

On the other hand, as far as we know, there has been no previous work about the adaptation

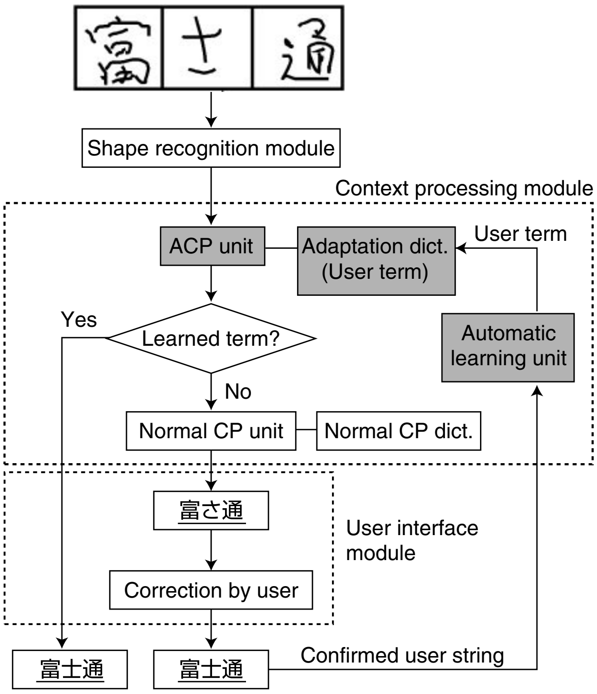


Figure 6
Adaptive context processing.

of context processing. However, if the correct character is included in the recognition candidates of the shape recognition module, good context processing can output the correct result. Therefore, we have developed an adaptive context processing (ACP) technology[14)] which improves the recognition accuracy by automatically collecting user's input strings and applying the information to the context processing.

Adaptive context-processing

Figure 6 shows the concept of our ACP mechanism. After processing by the shape recognition module, the ACP unit evaluates all possible strings generated by the recognition candidates using the adaptation dictionary. The adaptation dictionary stores previously input user terms (sub-strings) extracted by the automatic learning unit. If a valid user term is found in the adaptation dictionary, the context processing module outputs the term as the recognition result. If no valid user term is found, the normal context processing (currently character-class Bi-gram based CP) is performed. If the recognition result is not what the user intended, the user corrects the output by some means (e.g., candidate selection). Finally the automatic learning unit collects the confirmed string then extracts the terms and stores them in the adaptation dictionary. Once a term is stored in the adaptation dictionary, the OLCR outputs the correct result when the user writes the same string. As we can see from Figure 6, the user does not need to make an extra operation unless only the OLCR system is being used.

In the automatic learning unit, an input user string is separated into a series of terms by heuristic text splitting rules. (We do not use morpheme analysis so as to save memory and because the input text is sometimes too short to apply.) A new term is simply added and a term that has previously occurred updates its appearance count and its last-appearance date.

Let the input sequence of a handwritten pattern be $X = X_1X_2 \cdots X_n$ and the corresponding character sequence be $C = C_1C_2 \cdots C_n$. Then, we define the evaluation value $L(C\,|\,X)$ as follows:

$$L(C|X) = \Sigma \mathrm{Sim}(X_i, C_i) + P(C) ,$$

where

$P(C) > 0$ if C exists in the adaptation dictionary
$P(C) = 0$ if C does not exist in the adaptation dictionary
$\mathrm{Sim}(X_i, C_i)$ is the recognition score of character C_i corresponding to input pattern X_i.

The value $P(C)$ is calculated as a function of the length of the term, the appearance count of the term, and the time interval from the last-appearance date. In the automatic learning unit, some old and least-used terms are deleted to save memory and to avoid side effects from an accidental registration of invalid terms.

Table 5 shows the results of experiments we did to measure the effectiveness of the ACP. In these experiments we asked 14 subjects to write the same text (personal schedule data) three times. The first experiment was done without ACP (i.e., normal CP only) and the second and third experiments were performed with ACP. The table shows that our ACP is effective and significantly improves recognition accuracy, although the recognition accuracy before the context processing is almost the same during the three experiments.

By using adaptive context processing, we can avoid repeated incorrect recognitions. We think this is very important because such repeated incorrect recognitions would make the user irritated with the OLCR system. Not only does our ACP improve recognition accuracy but it also enhances user satisfaction. Compared to adaptation to shape recognition, our ACP requires much less memory and the management of the user dictionary is much easier.

We are sure that adaptation will be one of the key technologies of future OLCR systems. Also, we think that the *interactivity* of OLCR is what distinguishes it the most from OCR systems. The shape recognition algorithms of OLCR and OCR have recently started to merge with each other, but the interactive feature of OLCR systems will remain unique to OLCR systems. We therefore call our technology *Interactive Character Recognition*.

Table 5
Performance of adaptive context processing (ACP).

	1st test (without ACP)	2nd test (with ACP)	3rd test (with ACP)
1st recog. rate before CP	81.7%	81.8%	80.5%
1st recog. rate after CP (Final)	86.1%	89.1%	95.4%

Test data : personal schedule data (233 kanji + 167 non-kanji per subject)
Subjects : 9 male, 5 female (all working at Fujitsu Laboratories)

5. Conclusion

We have developed a high-performance Japanese OLCR system. To improve the recognition accuracy for complex kanji characters, we have developed a hybrid recognizer which integrates an online and an offline recognition module. In an experiment, our hybrid recognizer achieved an 86.8% recognition accuracy, while the online

Figure 7
Fujitsu's Pen-PC (FM PenNote model S1).

(a) TeamPad 7600 (b) TeamPad 7200

Figure 8
Fujitsu's TeamPad series of hand-held terminals.

and offline modules achieved scores of 84.3% and 72.4%, respectively. To discriminate between similar characters, especially non-kanji characters, we have developed a compact and effective context processing module based on a character-class Bi-gram. In an experiment, our context processing module improved the recognition accuracy from 82.7% to 90.5% for non-kanji characters and from 90.6% to 93.8% for kanji characters with only a 40 KB dictionary. Finally, to improve the recognition accuracy for a specific user, we have developed an adaptive context processing technology. By automatically learning a string that has been input by the user, our ACP module gradually improves its recognition accuracy without any extra operation by the user. In an experiment, we observed that the recognition accuracy was improved from 86.1% to 95.4% by applying the ACP.

Our OLCR software has been adopted in Fujitsu's pen-based PC, the "FM PenNote model S1" (**Figure 7**), which is a Japanese version of the Stylistics series of FPSI (Fujitsu Personal Systems Inc.). It has also been adopted in Fujitsu's TeamPad series of hand-held terminals (**Figure 8**). Also, it is used in a Fujitsu Japanese text-input software (i.e., Kana-to-Kanji conversion software) for PCs named OAK (**Figure 9**). In addition, there is a middle-ware library for Fujitsu's SPARC-lite series and FR series of MPUs which support our OLCR. We are now planning to apply our OLCR to Fujitsu's self-service terminals (multimedia kiosks).

Figure 9
Character retrieval by handwriting (OAK V7.0).

We are making continuous efforts to improve recognition accuracy, to expand the OLCR target platform, and to relax the restraints on the user interface of OLCR systems.

Acknowledgements

The authors are very grateful to Prof. Masaki Nakagawa of TUAT and Dr. Katsuhiko Akiyama for their continuous help in the development of our OLCR technology. We thank our superior, Mr. Haruo Akimoto, for his advice and continuous encouragement. We also thank our colleagues, Mr. Iwao Otsuka and Mr. Kenji Nakajima, who played big roles in the development of our OLCR systems.

References

1) T. Masui: An Efficient Text Input Method for Pen-based Computers. Proc. of CHI'98, pp.328-335, 1998.

2) D. Venolia and F. Neiberg: T-Cube: A fast, self-disclosing pen-based alphabet. Proc. of CHI'94, pp.265-270, 1994.
3) K. Perlin: Quikwriting: Continuous Stylus-based Text Entry. Proc. of UIST'98, pp.215-216, 1998.
4) H. Tanaka et al.: Hybrid Pen-Input Character Recognition System Based on Integration of Online-Offline Recognition. Proc. of ICDAR'99, pp.209-212, 1999.
5) M. Nakagawa and K. Akiyama: A linear-time Elastic Matching for Stroke Number Free Recognition for Online Handwritten Characters. Proc. of 4th IWFHR, pp.48-56, 1994.
6) K. Akiyama and M. Nakagawa: A Linear-time Elastic Matching Algorithm for On-line Recognition of Japanese Characters. (in Japanese), *Trans. of IEICEJ*, **81-D-II**, 4, pp.651-659 (1998).
7) K. Ishigaki and T. Morishita: A Top-down Online Handwritten Character Recognition Method via the Denotation of Variations. Proc. of Int. Conf. of Computer Processing of Chinese and Oriental Languages 1988, pp.141-145, 1988.
8) H. Kamada, K. Fujimoto, and K. Kurowaka: High-speed, High-accuracy Document Recognition System for Languages Containing Many Characters. Proc. of 3rd IAPR Workshop on Document Analysis Systems (DAS'98), pp.177-180, 1998.
9) T. Wakabayashi, Y. Deng, S. Tsuruoka, F. Kimura, and Y. Miyake: Accuracy Improvement by Nonlinear Normalization and Feature Compression in Handwritten Chinese Character Recognition. (in Japanese), *Trans. of IEICEJ*, **J79-D-II**, 5, pp.765-774 (1995).
10) M. Nakagawa et al.: Collection and utilization of on-line handwritten character patterns sampled in a sequence of sentences without any writing instructions. (in Japanese), IEICE Technical Report, PRU95-110, pp.43-48, 1995.
11) N. Itoh: A Method of Post-processing for Online Japanese Character Recognition based on Bigrams. (in Japanese), *Trans. on Natural Language Processing of IPSJ*, **93**, 79, pp.37-44 (1993).
12) O. Kato et al.: A Handwriting Input System for Japanese. Proc. of IFIP'80, pp.689-694, 1980.
13) Y. Kimura and K. Okada: Effectiveness of Personal Dictionaries in Handy Type Pen-Input Interface. (in Japanese), IEICEJ Technical Report, PRMU97-4, pp.25-32, 1997.
14) N. Iwayama: A Handwriting Character Recognition System with Adaptive Context Processing. (in Japanese), IPSJ Technical Report on Human Interface, 84-15, pp.85-90, 1999.

Kazushi Ishigaki received the B.S. and M.S. degrees in Information Science from the University of Tokyo, Tokyo, Japan in 1980 and 1982, respectively. He joined Fujitsu Laboratories Ltd. in 1982 and has been engaged in research and development of online handwriting character recognition and human interfaces. He is a member of the Information Processing Society of Japan (IPSJ) and the Human Interface Society (HIS) of Japan.

Hiroshi Tanaka received the B.S. degree in Communication Engineering from Tohoku University, Sendai, Japan in 1986. He joined Fujitsu Laboratories Ltd., Kawasaki, Japan in 1986 and was engaged in research and development of speech recognition systems until 1991. Since late 1991 he has been engaged in research and development of online handwriting character recognition. He is a member of the Acoustic Society of Japan (ASJ) and the Institute of Electronics, Information and Communication Engineers of Japan (IEICE).

Naomi Iwayama received the B.S. degree in Mathematics from Kyoto University, Japan in 1988. She joined Fujitsu Laboratories Ltd. in 1988 and became engaged in research and development of natural language processing and childrens' communication systems. Since 1998 she has been engaged in research and development of context post-processing of interactive character recognition. She is currently interested in intelligent user interfaces.