UDC 681.327

Application of Single Hand Keys Input Scheme to Pocket Computer

●Masakatsu Sugimoto

(Manuscript received August 10, 1999)

This paper describes an input scheme for a pocket computer. The scheme "Single Hand Keys (SHK) for mobile computers" has been applied to the Palm III pocket computer made by 3Com Corporation. In this paper, we explain our SHK KeySeg, which is a small keyboard designed for the Palm III and its support programs. Then we discuss the human-machine interface we have designed and implemented for the support programs. We will also touch on future activities for realizing the next generation of mobile information tools.

1. Introduction

Mobile computing requires information tools which can be used anytime and anywhere. Such information tools include Personal Digital Assistants (PDAs), portable terminals, pocket computers, and laptop computers. This paper introduces the Single Hand Keys (SHK) input scheme for mobile computing. The SHK input scheme uses only 18 keys and has no shift operation. With this scheme, a user can input English text and Japanese text at high speed. We have established input schemes based on dictionary searches for both English text and Japanese text. In the 21st century, SHK will be applied to miniaturized and portable information tools such as wallet-size computers.

First, we describe the background of SHK and the research steps that we have taken. Then we describe a small keyboard based on SHK and the SHK support programs for a pocket computer. The keyboard "SHK KeySeg" attaches directly to the Palm III made by 3Com Corporation and compatible pocket computers. The SHK KeySeg can be used in various situations, for example, it can be used to take a memo at high speed while sitting on a seat in a commuter train or while standing.

2. Background

Computers have become remarkably small and light. Desktop PCs and notebook PCs now have a much higher performance than the large computers of 20 years ago that occupied an entire room. In fact, so much progress has been made that, notebook PCs weighing less than one kilogram are now available.

When users operate computers at the office or home, they usually have to sit in a chair in front of a desk on which the equipment is placed (in this paper, this way of using computers is called "desk-work" use). Notebook PCs are operated in a similar fashion even though they are small enough and light enough to be carried around.

With the incredible pace of development in electronics technology, we can expect the eventual arrival of highly miniaturized computers. To identify the potential bottlenecks in miniaturized computers, we can speculate about the ways people will use computers when the size and weight of their microprocessor, memory, battery, etc., become negligible. When such a computer is perfected, people will be able to use it wherever they are and whatever they are doing, and the method of use could then truly be called "mobile."

The mobile era of computers is just beginning. Unlike desk-work use, which has a long history of research and development, mobile use will require much research and development in various areas.[1] A mobile computer should include functions for the following: document preparation, electronic mail, and Web browsing. In the future, mobile computers will be able to do almost everything that desktop PCs do plus many other things.

- Themes for mobile use

The input and output schemes for mobile computers are important research themes. Input through a full keyboard is reliable and fast. However, a full keyboard is not suited for mobile use because it is too large to carry and because it is difficult to type fast while standing.

There are several alternative input methods for mobile use; namely a small, full keyboard with tightly packed keys, voice recognition, and handwritten character recognition. However, touch typing is not practical on tightly packed keys. Also, in certain locations voice input is not practical, and the input speeds allowed by handwritten character recognition systems are much lower than speeds attainable by touch typing.

- One solution

On a full keyboard, we can input characters fast by typing with both hands (i.e., by touch typing). But we considered that typing speeds similar to full-keyboard speeds might be attainable through a single-handed scheme provided its keys could be typed at a similar speed. This led us to develop the SHK input scheme. Basically, we built a keyboard that was half the size of a full keyboard and then developed a scheme that enables high-speed input of English and Japanese text using a single hand.

3. Design targets for SHK

We designed SHK for persons who use mobile information tools daily for long periods, rather than for novice users. That is, SHK was designed for people who have been using a full keyboard for a long time and are competent at touch typing. We are also preparing methods to help beginners learn SHK typing.

Our other goals for SHK were that the scheme should be equally usable by right-handed and left-handed persons, and that it should enable input of every key code now available on a full keyboard.

4. Research steps

4.1 Input for English text

We began by designing an input scheme for English text. To realize high-speed input, the keys were arranged in three rows and six columns (18 keys in total) so that the user can touch type.[2), 3)] Since there are 26 characters in the western alphabet and SHK has only 18 keys, there is a shortage of keys. This problem is solved in SHK by giving A and E their own keys and assigning the remaining 24 characters in pairs (**Figure 1**). However, there is no shift function, so the user cannot immediately specify which of the two characters on a key is the intended one. Instead, after a word has been input, the user presses a special key named the Ambiguity resolution (Ar) key. To resolve an ambiguity, a support program collects the candidate words for the input string by searching an internal dictionary and then displays them. Usually, only one candidate exists for a given English input. For the inputs that have several candidates, the candidate words are displayed one at a time. To select the required word, the user presses the Ar key.

Each row has six keys. At the home position row (middle row), the index finger covers its home

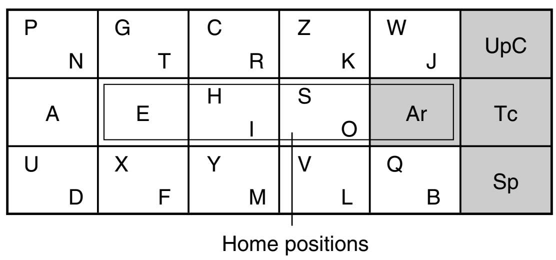

Figure 1
Key arrangement of SHK.

position key (E) and the key to the left (A). The little finger covers the Ar and Tc keys, and the middle finger and ring finger cover the H/I and S/O keys, respectively. The same finger assignment applies to the upper and lower rows. The details of how we decided on the arrangement of keys are discussed in Refs. 4) and 5).

We built a prototype front-end processor (FEP) which realized the SHK arrangement on Windows 3.1. Then we collected data about the operation of the SHK keyboard, for example, how easy its keys can be operated, what typing speeds could be achieved, and the organization of its internal dictionary.

When we started our research for SHK, we were using a keypad that was operated on a desk. However, we eventually realized that we could place all the keys on a small, thin device that the user could hold and operate with a single hand. This led to our developing the SHK keycard and the "castanets" method of operation. We formally demonstrated the castanets operation of the SHK keycard at the CHI '96 International Conference in April 1996.[2] This was the world's first demonstration of high-speed text input with one hand, and we received very favorable responses from the audience. After this event, we created a prototype software for Windows 95/NT.

4.2 Input for Japanese text

After we finished the input scheme for English text, we started to develop a Japanese input scheme for the SHK key arrangement. As a result, we came up with the idea of input through the use of "compressed Roman characters."[6]

In conventional input schemes for Japanese text, conversion from a Roman character string to a Kana string (phonetic character string) is widely used. The Kana string is then converted to a Kanji (Chinese character) string. SHK also uses Roman character strings. However, we call the Roman character strings input using the Japanese input system "compressed Roman character strings," because they mostly consist of ambiguous alphabetic pairs.

In SHK, the compressed Roman character strings are directly converted to Kanji strings. This is because, if we try to convert a compressed Roman character string to a Kana string, there are too many candidates to be handled. Because we usually do not include a separating space between words in Japanese text, the conversion logic becomes more complex than with word-by-word ambiguity resolution, which is applied to English text input. In Japanese text input, when the user presses the Ar key at the end of a word, idiom, or phrase, the internal logic starts the ambiguity resolution.

5. SHK KeySeg and support programs for pocket computers

While we were making the prototype systems by combining the SHK keys and support programs on a PC, several computers which can be classified as "almost wallet-size" became available on the market. Those pocket computers were then improved in terms of processing speed and memory size. Among them, the Palm Pilot and the Palm III from 3Com Corporation were particularly interesting.

We selected the Palm III as a base computer for a prototype for the following reasons.

Because our long term target was to realize a wallet-size computer which could be used by a wide range of people, we wanted to develop a prototype system that was as close to wallet-size as possible. We set the following conditions for our prototype pocket computer.

– It must be small enough to hold with one hand and operate with the other hand.
– It must be easy to attach the SHK keys.
– It must be easy to develop support programs.

The Palm III had a suitable size and weight and we were able to attach a keyboard via its com-

Note: "Palm III" refers to both the Palm Pilot and the Palm III in the following text unless otherwise specified.

munication port. For the software development, CodeWarrior from Metrowerks Corporation and GCC were available. We therefore decided to develop a prototype SHK input system on the Palm III. The prototype system includes a new SHK keyboard for the Palm III and support programs. We named the SHK keyboard attached to the Palm the "KeySeg."

Figure 2
SHK KeySeg.

Figure 3
Using the SHK KeySeg.

5.1 Major features of prototype system

The major features of the SHK KeySeg for the Palm III are as follows:

1) SHK KeySeg

- Small keyboard which can be touch-typed
- Right-handed and left-handed key arrangements
- Built-in slide-out stand
- Low weight and low battery power consumption

2) Support programs

- FEP type support programs
- Registration of user's words

Each feature is explained below in more detail.

5.1.1 SHK KeySeg

Figure 2 shows the SHK KeySeg. The size is 95 mm (length) × 80 mm (width) × 20 mm (height). The user holds it with one hand and operates it with the other, as shown in **Figure 3**. An experienced user can touch type on the KeySeg.

We have prepared key arrangements for right-handed and left-handed persons. The KeySeg shown in Figure 2 is for right-handed persons. In the KeySeg for left-handed persons, the keys are simply left-right transposed from their positions in the right-handed version. Figure 3 shows one way of using a Palm III with the KeySeg. Another way is to put it on a desk with the built-in slide-out stand as shown in **Figure 4**.

Figure 4
Use with built-in slide-out stand.

The KeySeg weights only 75 g. The power is supplied from the Palm III. It consumes about 1.0 mA in key input mode and about 0.01 μA in sleep mode. Usually the KeySeg is in sleep mode. When it receives a signal from a key, it enters input mode, processes the signal, and then immediately returns to sleep mode.

5.1.2 Support programs

To enable the user to input English and Japanese text into the various applications that can be run on the Palm III, we designed and implemented an FEP for the KeySeg interface. The FEP uses DaggerWare's Hack Master Extension feature, which is a kind of interrupt processing function.

The SHK support programs have an internal dictionary of about 10 000 words. If the word the user wants to use is not in the dictionary, the user can register it in the user's dictionary and then use it.

5.2 English text input

The SHK input scheme is best suited for input of the English words for which there are entries in the internal dictionary. These words are handled by a word-by-word Ar scheme. On the other hand, a character-by-character input scheme is applied for words which have no entry in the internal dictionary. We will now explain how the user inputs words, terminating characters, numeric characters, etc., in SHK.

5.2.1 Word-by-word Ar scheme

To input a word, the user presses the keys which correspond to the alphabetic characters in the word without indicating which of the two characters on each key (except the A and E keys) are required. The user then presses the Ar key, and a word is displayed on the screen. If this word is the intended one, the user inputs the next word and an intervening space character is automatically inserted. If the word is not the intended one, the user presses the Ar key repeatedly until the intended word is displayed. When a blank is displayed, there are no more alternative words. Pressing the Ar key redisplays the alternative words that have already been displayed.

5.2.2 Character-by-character input scheme

For the words which have no entry in the internal dictionary, the user must use the character-by-character input scheme. For example, the character-by-character scheme is used to input a new abbreviation. In this scheme, the user must distinguish between the paired alphabetic characters. Alphabetic characters can be input as follows:

– By pressing the keys for the following characters:
  N, T, R, K, J, A, E, I, O, D, F, M, L, B, or
– By pressing the Tc key followed by the keys for the following characters:
  P, G, C, Z, W, H, S, U, X, Y, V, Q.

After all the alphabetic characters of a word have been input, the user presses the Ar key.

5.2.3 Extra post-fixed space character

In English text, there is usually a space character between words. In the SHK input scheme, a space character is automatically post-fixed after the selected word, and the cursor is placed one position to the right of the space character.

5.2.4 Upper case generation

When the user presses the UpC key immediately after a word has been selected, the system enters into a mode in which three different case variations of the word are sequentially displayed, for example, "holiday," "Holiday," "HOLIDAY." The correct case variation is selected by pressing the Ar key.

5.2.5 Entering a period, comma, question mark, etc.

Various terminating characters (i.e., symbol characters and control characters) are needed for English text input. There is a special consideration in the SHK input scheme for the characters of this type which appear frequently in English text input. Symbol characters such as the period, comma, and question mark and control characters such as the back space, carriage return (or

Table 1
Shortcuts for punctuation characters.

Character	Input	Meaning
.	PE	Period
,	CO	Comma
:	COL	Colon
;	SE	Semicolon
?	QU	Question mark
!	XE	Exclamation mark
-	MI	Minus
"	QO	Quotation mark
'	TE	Apostrophe
cr	EE	Enter
<-	EA	Back space
<-<-	AE	Back space × 2

3	6	9	#	,	Bs
2	5	8	0	AR	Tc
1	4	7	*	.	Sp

Figure 5
Key positions of characters in num-mode.

%	{	}	"	`	^
@	$	!	'	Rsv	:
?	\	_	~	t. s.	Rsv

Note: t. s. : Time stamp
Rsv : Reserved

Figure 6
Key positions of terminating characters (Tc-UpC).

&	[	]	=	,	Rsv
+	(	)	*	Ca	|
–	<	>	/	.	;

Note: Ca : Cancel

Figure 7
Key positions of terminating characters (Tc-Sp).

enter) can be input using a method similar to the word-by-word Ar method or through a shortcut. For example, a period can be input by pressing P/N E and the Ar key, and a comma can be input by pressing C/R, S/O, and the Ar key. These shortcuts are shown in **Table 1**.

Also, there are ways of inputting terminating characters by changing the input mode. We will describe these later.

5.2.6 Numeric characters

Numeric characters and their related symbols can be input using two methods.

1) **Num-mode**
The user enters num-mode (numeric mode) to input one or more numeric characters or symbols and then exits num-mode. To enter num-mode, the user presses the Tc key twice. To exit num-mode, the user presses the Tc key twice or presses the Ar key once. **Figure 5** shows the key positions of characters in num-mode. The characters needed for telephone dialing are entered in this mode.

2) **Use of Ar key**
The user presses the required key according to the arrangement shown in Figure 5 and then presses the Ar key twice.

5.2.7 Terminating characters and symbol characters

Terminating characters can be input using two methods:

– Press Tc UpC and the required key according to the arrangement shown in **Figure 6**.
– Press Tc Sp and the required key according to the arrangement shown in **Figure 7**.

5.2.8 Covering variations of words

Variations of English words are created by adding a prefix or suffix or by using a different word ending. For example, to make the negative form, “familiar” can be changed to “unfamiliar” and to change from an adjective to the noun form, “bright” can be changed to “brightness.” Also, suffixing with “ed” makes the past tense of some verbs, and suffixing adjectives with “ly” often produces an adverb. There are two ways to handle

Table 2
Input of word elements.

Word element	Input	Example
ed	ED	visited
ing	ING	visiting
'll	LL	he'll
ly	LY	timely
n't	NT	haven't
're	RE	you're
's	DE	Fujitsu's
've	DV	they've

these variations.

1) **Word analysis method**
The user inputs all the characters of a word with the required prefix, suffix, and variation of word ending included. Then, when the user presses the Ar key, the internal logic processes the word using the internal dictionary and the stored rules for English words. For example, to input "friendly," the user inputs "FRIENDLY" and then presses the Ar key.

2) **Concatenation of words and word elements**
The user can make a word by concatenating two or more words or word-elements. For example, "friendly" can be input by inputting "friend" and then "ly."
For convenience, when the user types one of the character strings shown in the input column of **Table 2**, the cursor moves back one position to nullify the space character that was automatically post-fixed, then the corresponding word element is inserted.

5.2.9 Cursor movement using keys

The cursor position on the screen can be changed by placing a stylus pen or a fingertip on it. However, the user may sometimes find it difficult to position the cursor precisely with this method, so the following method of controlling the cursor is provided. The Tc key is pressed three times, then:

Cursor left : E key
right : S/O key
up : C/R key
down: H/I key.

5.2.10 Input speed

Once a user becomes familiar with SHK, input speeds of 40 to 60 words per minute are possible; these speeds are comparable with speeds achieved with a full keyboard. A high input speed is possible because pressing the Ar key at the end of a word usually gives only one candidate word and because a space character is post-fixed to the word automatically.

5.3 Japanese text input

We have developed support programs that convert compressed Roman character strings to Japanese text in units of words, idioms, or phrases. The major points of Japanese text input are explained below.

5.3.1 Roman character string to Kana correspondence

The correspondences between Roman character strings and Kana in SHK are the same as those used in ordinary Japanese word processors and text editors. These correspondences are given by two methods: the Kunrei method and the Hepburn method (after James Curtis Hepburn, who invented the method). In the Kunrei method, し is represented as "si," ち as "ti," and つ as "tu." Whereas, in the Hepburn method, し is represented as "shi," ち as "chi," and つ as "tsu." We use the Hepburn method for Japanese text input because it is widely used in everyday life in Japan.

However, although we use the Hepburn method, there are a few exceptions. We represent じ, as "zi" not "ji," and we represent じゃ, as "zya" not "ja." This is because we wanted to minimize the number of times the little finger needs to be moved from its home position. **Table 3** shows the Kana to Roman character correspondence used in SHK.

To input Japanese text, for example, "私は" (I am), the user types "WA-TA-KU-SHI-HA" and

あ	い	う	え	お	A	I	U	E	O
か	き	く	け	こ	KA	KI	KU	KE	KO
さ	し	す	せ	そ	SA	SHI	SU	SE	SO
た	ち	つ	て	と	TA	CHI	TSU	TE	TO
な	に	ぬ	ね	の	NA	NI	NU	NE	NO
は	ひ	ふ	へ	ほ	HA	HI	FU	HE	HO
ま	み	む	め	も	MA	MI	MU	ME	MO
や		ゆ	いぇ	よ	YA		YU	YE	YO
ら	り	る	れ	ろ	RA	RI	RU	RE	RO
わ	うぃ		うぇ	を	WA	WI		WE	WO
ん					N				
が	ぎ	ぐ	げ	ご	GA	GI	GU	GE	GO
ざ	じ	ず	ぜ	ぞ	ZA	ZI	ZU	ZE	ZO
だ	ぢ	づ	で	ど	DA	DI	DU	DE	DO
ば	び	ぶ	べ	ぼ	BA	BI	BU	BE	BO
ぱ	ぴ	ぷ	ぺ	ぽ	PA	PI	PU	PE	PO
きゃ	きぃ	きゅ	きぇ	きょ	KYA	KYI	KYU	KYE	KYO
しゃ	しぃ	しゅ	しぇ	しょ	SHA	SYI	SHU	SHE	SHO
ちゃ	ちぃ	ちゅ	ちぇ	ちょ	CHA	CYI	CHU	CHE	CHO
にゃ	にぃ	にゅ	にぇ	にょ	NYA	NYI	NYU	NYE	NYO
ひゃ	ひぃ	ひゅ	ひぇ	ひょ	HYA	HYI	HYU	HYE	HYO
ふぁ				ふぉ	FA				FO
ふゃ	ふぃ	ふゅ	ふぇ	ふょ	FYA	FYI	FYU	FYE	FYO
みゃ	みぃ	みゅ	みぇ	みょ	MYA	MYI	MYU	MYE	MYO
りゃ	りぃ	りゅ	りぇ	りょ	RYA	RYI	RYU	RYE	RYO
ぎゃ	ぎぃ	ぎゅ	ぎぇ	ぎょ	GYA	GYI	GYU	GYE	GYO
じゃ	じぃ	じゅ	じぇ	じょ	ZYA	ZYI	ZYU	ZYE	ZYO
ぢゃ	ぢぃ	ぢゅ	ぢぇ	ぢょ	DYA	DYI	DYU	DYE	DYO
びゃ	びぃ	びゅ	びぇ	びょ	BYA	BYI	BYU	BYE	BYO
ぴゃ	ぴぃ	ぴゅ	ぴぇ	ぴょ	PYA	PYI	PYU	PYE	PYO
つぁ	つぃ		つぇ	つぉ	TSA	TSI		TSE	TSO
てゃ	てぃ	てゅ	てぇ	てょ	THA	THI	THU	THE	THO
でゃ	でぃ	でゅ	でぇ	でょ	DHA	DHI	DHU	DHE	DHO
ヴぁ	ヴぃ	ヴ	ヴぇ	ヴぉ	VA	VI	VU	VE	VO
ぁ	ぃ	ぅ	ぇ	ぉ	LA	LI	LU	LE	LO
ゃ		ゅ		ょ	LYA		LYU		LYO

Table 3
Kana-Roman character correspondence of SHK.

then presses the Ar key (The "-" symbol is used here in this paper to delineate the syllables for the reader and are not actually input). Then the system displays the Japanese text string "私は." Since the user does not distinguish between the two alphabetic characters on each key, the actual keying is "W/J-A-G/T-A-Z/K-U/D-S/O-H/I-H/I-H/I-A."

5.3.2 Difference from conventional Roman-Japanese conversion systems

In conventional word processors, when the user is inputting alphabetic characters and then converting them to Kana, the strings are converted immediately after the conversion rule can be applied. Therefore the user sees the Kana characters as the alphabetic characters are input.

In SHK, however, the user sees only a compressed alphabetic string until the Ar key is pressed. This looks like a weak point of SHK. However, after a while, users become so familiar with the input patterns that they can type without needing to look at the screen to check each character before pressing the Ar key.

5.3.3 Increasing the number of alternative words

In SHK, two alphabetic characters are allocated to each key except for the A and E keys. The keys which generate the most alternative words are the N/P, T/G, C/R, K/Z, H/I, S/O, D/U, and M/Y keys. For example, for "KA-I-DA-N," there are two candidate Kana strings: "Kaidan" (階段 = stairs) and "Zaidan" (財団 = foundation).

The number of candidate words for SHK tends to be larger than in conventional systems. In the support programs for the Palm III, the mechanism of the personal vocabulary space described in Ref. 3) has not been implemented. However, we are implementing it for the next versions of the support programs.

5.3.4 Inputting terminating characters and numeric characters

As is the case for English text input, the user can use shortcuts for the terminating characters, for example, the shortcut for a period is "P/N-E-Ar." A user can also input numeric characters and symbols by changing the input mode.

5.3.5 Seamless change between Japanese and English text input modes

It is sometimes necessary to insert an English word or phrase into Japanese text. To do this, easy changing between Japanese text input and English text input would be helpful. In SHK, there are two ways to change the mode. One is to set a check mark on a control panel, which is the usual way. The other way, using keys, is unique to SHK.

To enter English mode, the user types "l-ei" and presses the Ar key ("ei" is for Eigo [English]).

Conversely, when in English mode, to enter Japanese mode, the user types "l-ja" and presses the Ar key("ja" is for Japanese). In our evaluations, this method of changing input modes is acceptable to users.

5.4 Fingertip operation and keys for coming wallet-size computers

The human interface design for the Palm III was centered on using a stylus pen on a screen. Almost all the screen operations performed by the user are done by a stylus pen. By attaching the SHK KeySeg to the Palm III, the user can input using the KeySeg keys. For other operations such as menu selection and button pushing, we found that users could do without a stylus pen. Users can specify the operations using their fingertips, provided that the menus and buttons are carefully designed and implemented. Operation by keys and fingertips only is very convenient because there is no need to pick up the stylus pen.

We believe that, in the future, the input operations for the smaller mobile tools such as wallet-size computers, will be done without a stylus-pen. We think this will be one of the main design points concerning the human-interface of the coming wallet-size computers.

6. SH-Keys

We exhibited and demonstrated English text input using the SHK prototype system on a Palm III at COMDEX, November 1998. The system attracted lots of people who were interested in input schemes for mobile information tools. Then, in March 1999, Fujitsu Takamisawa Component Ltd. began national sales of the SHK KeySeg for the Palm III and various support programs. SH-Keys works on the Palm Pilot, Palm III, and compatible computers, including IBM's WorkPad.

One of the users who started to use the KeySeg and the support programs for Japanese input told us he was able to master touch typing on the SHK KeySeg in a week and that he was carrying it and the Palm III with him and using it wherever he went.

There are keyboards that can be attached to the Palm III, for example, the GoType keyboard and the Newton keyboard. These are full keyboards that are small and light. However, it is difficult to touch type on them while standing; whereas, with the KeySeg, a user can easily touch type while standing.

7. Further work

Currently, the SHK input scheme can be used only with the Palm III. We would like to expand the areas of usage and are considering two ways to realize this:

- Writing an FEP for each OS used on mobile machines
- Making an SHK front-end hardware device for various mobile machines.

The first approach is a conventional one. We could write support programs for the KeySeg for each OS. However, although most of the code for each realization would be similar it would require a lot of effort. For the second approach, we could make an SHK front-end device having SHK keys, a microprocessor, and memory which performs the main functions of the front-end processor. The SHK front-end device would be connected to another machine through a wired or wireless communication channel.[7)]

When we compare these two approaches, the second one appears much more attractive than the first. In the second approach, the benefits would mostly be felt by the users. The users only need to carry one SHK front-end device. The device can be used with whatever information tools the user wants, for example, a wallet-size computer, pocket computer, PDA, PC, or even a workstation. If we extend this idea, the SHK front-end device would be useful as a text input component of office tools such as electronic dictionaries, handy electronic language translators, and even the home appliances of the future, if they need text data. To realize an SHK front-end device, the following should be researched.

- A standardized interface between the front-end device and the host-machine that needs text data
- The logic for the front-end device and the host-machine.

We are sure that this idea can be extended from a text-based front-end device to knowledge-based front-end devices and to multimedia data front-end devices.

8. Conclusion

This paper has introduced the SHK input scheme for mobile information tools of the near future, for example, wallet-size computers. Then, we described the SHK KeySeg and support programs for English text input and Japanese text input to the Palm III. Using the SHK KeySeg, users can input text at high speed even while standing.

A mobile information tool with fingertip/key-touch operation and no stylus pen will be the next target. Realization of an SHK front-end device will be another challenge. We will continue our research of mobile information tools and their related infrastructure for the next stage of mobile computing.

References

1) M. Tsukamoto: Recommendations in the Mobile Computing Research. (in Japanese), *Journal of IPSJ*, **39**, 2, pp.158-162 (1998).
2) M. Sugimoto and K. Takahashi: SHK: Single Hand Key Card for Mobile Devices. Conference Companion, CHI'96, pp.7-8 (1996).
3) M. Sugimoto: Single-Hand Input Scheme for English and Japanese Text. *FUJITSU Sci. Tech. J.*, **33**, 2, pp.189-195 (1998).
4) M. Sugimoto: Single-handed Keyboard having Keys with Multiple Characters and Character Ambiguity Resolution Logic. US Patent 5,847,697, Dec. 8, 1998.
5) M. Sugimoto: Input Scheme for Japanese Text with SHK Keycard. (in Japanese), Workshop on Mobile Computing, Information Processing Society of Japan, May 30, 1-1, 1997.
6) M. Sugimoto: Application to the pocketable computer of SHK (Single Hand Keys) Input Scheme. (in Japanese), Workshop on Mobile Computing, Information Processing Society of Japan, No.5, May 29, 1998.
7) B. Myers : An Implementation Architecture to Support Single-Display Groupware, CMU-CS-99-139, May, 1999.

Masakatsu Sugimoto received the B.S., M.S., and Dr. degrees in Electronic Engineering from the University of Tokyo, Japan in 1965, 1967, and 1970, respectively. He has been working for Fujitsu for about 30 years. He has written papers in various fields, for example, logic design automation systems, language-directed computer design, software engineering, machine translation systems, the Japanese fifth-generation computer system, and knowledge bases. His current research interest is mobile computing. He is a member of the IPSJ, JSAI, JSSST, IEEE, and ACM. He received the Yamashita Research Award from the IPSJ in 1999.