Arranging Touch Screen Software Keyboard Split-keys based on Contact Surface

Kentaro Go
Interdisciplinary Graduate School
of Medicine and Engineering
University of Yamanashi
4-3-11 Takeda, Kofu 400-8511
Japan
go@yamanashi.ac.jp

Leo Tsurumi
Interdisciplinary Graduate School
of Medicine and Engineering
University of Yamanashi
4-3-11 Takeda, Kofu 400-8511
Japan
tsurumi@golab.org

Acknowledgements
Part of this work was supported by FY2009 Research for Promoting Technological Seeds (A), Japan Science and Technology Agency.

CHI 2010, April 10–15, 2010, Atlanta, Georgia, USA.
ACM 978-1-60558-930-5/10/04.

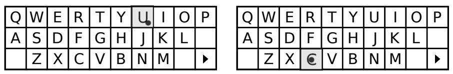
Abstract
Touch screen devices, which have become ubiquitous in our daily lives, offer users flexible input and output operations. Typical operation methods for touch screen devices include the use of a stylus or a finger. A touch screen user can select a stylus or finger depending on the user's situation and preference. In this paper, we propose a dynamic method of assigning symbols to keys for a software keyboard on a touch screen device. This method provides flexible adjustment to both the stylus operation and finger operation.

Keywords
Text entry, mobile device, touch screen, split key

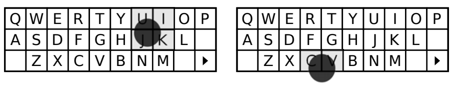
ACM Classification Keywords
H5.2. Information interfaces and presentation (e.g., HCI): User Interfaces.

General Terms
Design, Human factors, Performance

Introduction
Portable computing devices with a touch screen such as mobile phones, personal digital assistants, and portable music players have become familiar devices. Using a touch screen as an input–output device can enable the

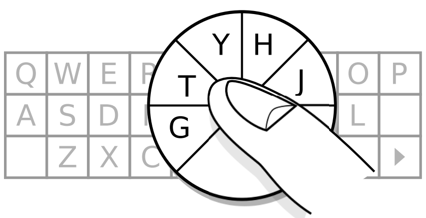


(a) Stylus operation examples: Left, the U key selection; Right, the C key selection.



(b) Finger operation examples: Left, the U, I, J, or K key selection; Right, the C or V key selection.

figure 1. Dynamic split keys in use. The circles on the keyboards represent user's contact surfaces.



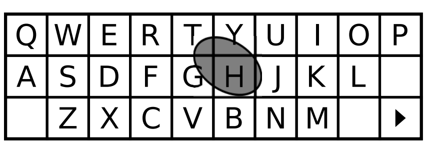


figure 2. Text entry option of the pie menu: Top, example of the pie menu visual feedback; Bottom, user's contact surface.

body of a portable computing device to be smaller and can make the look-and-feel of the user interface much more intuitive. Flexible input operations are possible, such as gesture commands, freehand input, and tapping and dragging of on-screen objects.

A typical and critical task for a touch screen is text entry. Major commercial touch screen devices have a software keyboard for text entry. It is assumed either that a stylus is used for smaller devices such as PDAs or that a finger is used for larger devices such as tabletops and interactive surfaces. Furthermore, software keyboards have various designs. It is assumed that a standard telephone keypad is used for smaller devices and that several letters are assigned to a key to use the limited screen space effectively. For larger devices, a standard qwerty keyboard is employed.

However, input device and keyboard design based on the screen size ignores the user's circumstances and preferences. Users might prefer to use fingers for short intermittent use in replying to an SMS message or micro-blogging even if typing with a software keyboard with a stylus is usually used.

As described in this paper, we propose a touch screen keyboard that dynamically arranges split-keys based on the contact surface area and position. Based on the contact surface, the dynamic split-key software keyboard adjusts to a user's circumstances. It addresses occlusion and fat finger problems of finger use on the touch screen.

Related Works

Several works have examined assignment of multiple letters to a single key. For example, a qwerty-like-phone keypad [1] uses a qwerty-like key pattern for a standard telephone keypad. Alphabetically Constrained Design [2] uses complete enumeration and Genetic Algorithm-based heuristic to find keypad design (mappings of letters to keys) solutions. It creates several unconstrained and constrained keypad designs. Stick [3] and TouchMeKey4 [4] have more sophisticated keypad design with less-standard keypads. MacKenzie and Soukoreff [5], after conducting extensive surveys of mobile text entry, presented a theoretical discussion and proposed the key-ambiguity continuum. Other examples of keypad design are examined in that study. In addition, a universal technique like Shift [6] can be used for typing small on-screen keyboard.

Prior studies of keypad design use either a qwerty arrangement or alphabetical order as a basic design. The former is used because of its property of transformation. Most users have some experience using a qwerty standard keyboard for PC use. For that reason, the qwerty-based keypad enables ease of learning. Alphabetical order is used for its reasonability. Users with less experience on a qwerty keyboard can comprehend it.

Dynamic Split Keys

Software keyboards' sizes and shapes depend on the touch screen and the device used with it, e.g. a stylus or bare finger. A stylus can point and select smaller a target on a screen. Consequently, a standard qwerty keyboard, with one key pressed per touch operation, might be appropriate. A bare finger can point and select a larger target on a screen: a split-key keyboard with good use of screen space might be appropriate.

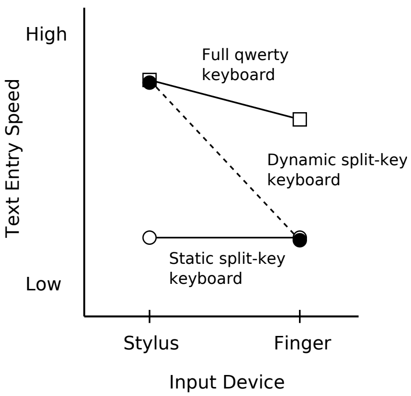


figure 3. Hypothetical text entry speed (Top) and error rate (Bottom).

Basic concept: dynamic split keys

This paper presents a proposal of a dynamic split-key software keyboard that adjusts to stylus and bare finger operation, providing advantages of both a full qwerty keyboard and a split-key keyboard for a touch screen. Figure 1 presents the dynamic split-key keyboard concept. Fundamentally, the contact surface size and position change visual feedback and selection processes. With a small contact surface, as in area stylus operation (figure 1(a)), it behaves as a standard software keyboard with no key selection ambiguity. With a large contact surface, as in finger operation (figure 1(b)), it behaves as a split-key keyboard with key selection ambiguity. The ambiguity is resolved using techniques such as menu selection, callout, and multi-tap. An example is to ask the user to select one display item from a pie menu, as figure 2 shows.

The dynamic split-key software keyboard presents various advantages for touch screens. First, it deals with differences among individuals, such as different finger sizes and contact surfaces. Our approach adapts to any finger size and contact surface. Second, it deals with differences among use contexts. A user might operate it using one-handed thumb use while others use it with two hands, holding the body of device with the non-dominant hand and operating it using the dominant hand. It therefore creates different contact surfaces. Third, it addresses personal preferences for a larger software keyboard and preferences for a smaller one. A user might even prefer the pie menu or flick-input method as a key selection method. Finally, this method is adaptable to different display sizes, from the tiny touch screens of mobile phones to the large touch screens of tabletop and interactive surfaces.

Moreover, our approach can use any selection method for input letter disambiguation. As described in this paper, we adopt the pie menu as the selection method (figure 2) because all selection items are accessible with a stylus or finger with minimum sliding operation.

Hypothesis related to performance

Regarding the performance of the dynamic split-key software keyboard for a touch screen, we present the following hypothesis.

Hypothesis: The dynamic split-key software keyboard for a touch screen in stylus use has the same performance as a full qwerty keyboard; in finger use, it has equal performance to a static split-key keyboard.

A hypothetical text entry speed is presented in figure 3(Top). In stylus use, a full qwerty software keyboard has the best performance on text entry speed among the three keyboard types. Using the full qwerty software keyboard, each letter can be typed using a single operation. For instance, a user points and selects the appropriate key k when she wishes to type k. In contrast, a static split-key keyboard has the worst performance on text entry speed among the three keyboard types. Using the static-split key keyboard, each letter must be typed with more than single operation. A user wishing to type k must point and select a group key representing m, k, and l. Then she resolves the ambiguity of keys m, k, and l by selecting k from the displayed menu items. Finally, the dynamic split-key software keyboard for a touch screen behaves as a full qwerty software keyboard to provide the same text entry speed as a full qwerty keyboard.

q	w	e	r	t	y	u	i	o	p
a	s	d	f	g	h	j	k	l	
	z	x	c	v	b	n	m		▶

figure 4. Keyboard design for the full qwerty keyboard and the dynamic split-key keyboard.

qaw	edr	tgy	uji	olp
sz	xfc	vhb	nkm	▶

(a) Large and medium size

qawd	rtgyu	jiolp
zsxc	fvbh	nmk▶

(b) Small size

figure 5. The static split-key keyboard design.

With finger use, the full qwerty software keyboard has the best performance for text entry speed among the three keyboard types, even though its performance is slightly worse: the finger use with the touch screen causes the occlusion problem and fat finger problem. The static split-key keyboard has the worst performance of text entry speed among the three keyboard types; nevertheless, its performance maintains the same text entry speed as that of stylus use. Finally, the dynamic split-key software keyboard for the touch screen behaves like the static split-key keyboard, offering the same text entry speed as that of the static split-key keyboard.

Figure 3(Bottom) presents the hypothetical error rate. In stylus use, the three keyboard types maintain similar error rates because the stylus use enables finer operations. Among the three keyboard types, the static split-key keyboard has the best error rate performance: it has larger key sizes. A full qwerty keyboard has the worst error rate performance among the three keyboard types because of its smaller key sizes. Finally, the dynamic split-key software keyboard for a touch screen behaves like the full qwerty software keyboard to have the same error rate as a full qwerty keyboard.

In finger use, the static split-key keyboard has the best error rate performance among the three keyboard types because the occlusion problem and fat finger problem on the finger use of touch screen are resolved using larger group keys and menu selection. The full qwerty keyboard has the worst error rate performance among the three keyboard types. The occlusion and fat finger problems greatly affect the smaller key size of the full qwerty keyboard. Finally, the dynamic split-key software keyboard for a touch screen behaves as a static split-key keyboard, showing the same error rate as the static split-key keyboard.

Evaluation

To test our hypothesis, we implemented a dynamic split-key software keyboard on a touch screen PDA and conducted evaluation. We used the pie menu to resolve the candidate letter ambiguity.

Evaluation task and participant

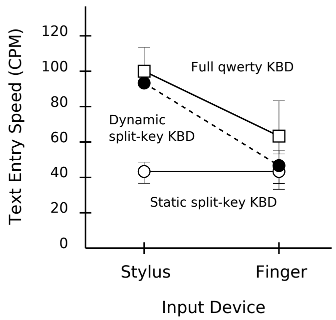
Our software keyboard evaluation is based on text entry speed and the error rate. Showing a random target phrase on the screen, we ask participants to type it using each software keyboard. We prepared 80 English proverbs as target phrases. The numbers of letters in the target phrases were 10–43 including spaces (28.5 letters on average).

We recruited six participants (two female, four male) from our Computer Science Department. The undergraduate students, who had much experience in typing with a standard qwerty keyboard on a PC, were instructed to proceed “quickly and accurately” while performing evaluation tasks. They were asked to continue typing without correction if they made errors.

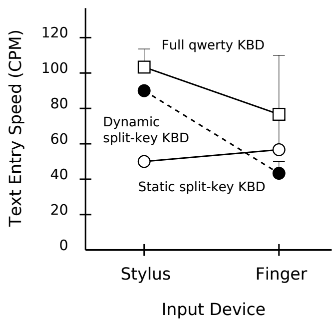
Experimental setting

Independent variables are keyboard type (full qwerty keyboard, static split-key keyboard and dynamic split-key keyboard) × key size (small, medium and large) × device (stylus and finger). The dependent variables are text entry speed and the error rate.

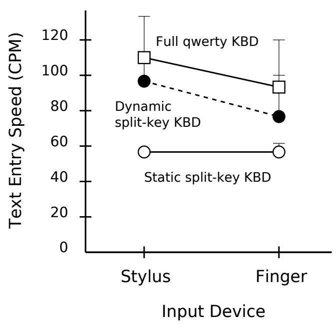
We used an HP iPAQ hx4700 Pocket PC, with a 4-inch touch screen detecting both stylus and finger use; it had a 640 × 480 pixel (VGA) 64K color TFT LCD.



(a) Small keyboard



(b) Medium keyboard



(c) Large keyboard

figure 6. Mean text entry speed.

Figures 4 and 5 show software keyboards presented on the screen. The full qwerty keyboard key size and the dynamic split-key keyboard are designed based on the key size of a standard PDA software keyboard (4 mm × 4 mm) as the medium size. We prepared the larger key size (5 mm × 5 mm) and the smaller size (3 mm × 3 mm). Therefore, the keyboard sizes were 50 mm (W) × 15 mm (H) for the large size, 40 mm × 12 mm for the medium size and 30 mm × 9 mm for the small size.

The static split-key keyboard has two designs. The large and medium keyboards have 10 keys (figure 5(a)). The small keyboard has six keys (figure 5 (b)). Their key arrangement resembles the standard qwerty keyboard. Ambiguity is resolved by displaying a pie menu and asking the user to select one menu item. To examine the basic performance and to test our hypothesis specifically, we assigned lower-case letters and the space key to the keyboards.

In the finger operation condition, each participant held the PDA with both hands and used the thumbs to type using the onscreen software keyboard. Both arms remained on the table to avoid fatigue. We asked participants to maintain the same holding style and body position during the experiment. The presentation order of the device (stylus and finger) is counterbalanced. Using the three keyboards, each participant conducts practice sessions immediately before she inputs 10 phrases chosen randomly for the actual experiment session.

Results and Discussion

Text entry speed

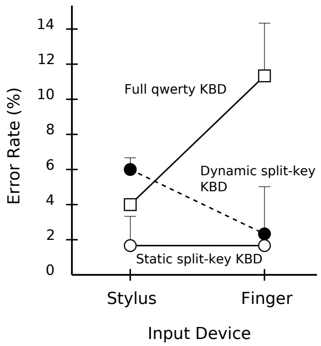
The text entry speed is calculated as characters per minute (CPM). Figures 6(a), (b), and (c) respectively show text entry speeds for small, medium, and large keyboards. Data were analyzed using ANOVA. The results are described based on a 5% significance level.

Using the small keyboard, the text entry speed of the static split-key keyboard was significantly lower than the other two keyboards in stylus use. For finger use, however, no significant difference was found among the three keyboards. Using the medium keyboard, the text entry speed of the static split-key keyboard was significantly lower than the other two keyboards in stylus use. For finger use, however, no significant difference was found among the three keyboards. For the large keyboard, the full qwerty keyboard text-entry speed was significantly higher than that of the static split-key keyboard for both stylus use and finger use. The dynamic split-key keyboard text-entry speed was significantly higher than that of the static split-key keyboard in stylus use. For finger use, no significant difference was found among the three keyboards.

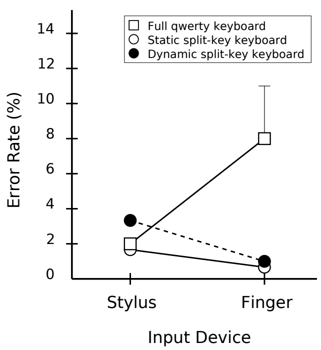
Error rate

The error rate is the number of incorrect letters among all input letters, represented as a percentage (%). Figures 7(a), (b), and (c) respectively portray error rates for small, medium, and large keyboards.

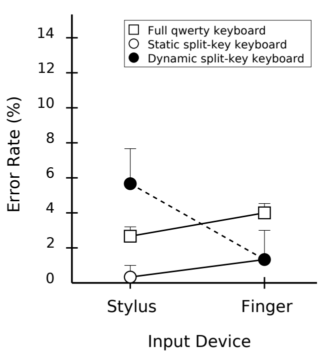
The small keyboard shows a significantly higher error rate for dynamic split-key keyboards than for the static split-key keyboard in stylus use. The full qwerty keyboard error rate was higher than the other two keyboards in finger use. For the medium keyboard, no significant difference was found among the three keyboards in stylus use. The error rate of the full qwerty keyboard was higher than those of the other two keyboards in finger use. For the large keyboard,



(a) Small keyboard



(b) Medium keyboard



(c) Large keyboard

figure 7. Mean error rate.

the error rate of the dynamic split-key keyboard was higher than the static split-key keyboard in stylus use. No significant difference was found among the three keyboards in finger use.

Discussion

The result supports our hypothesis that the dynamic split-key software keyboard for a touch screen in stylus use has the same performance as the full qwerty keyboard. The text entry speed of the dynamic split-key keyboard is significantly higher than that of the static split-key keyboard; no significant difference was found among the full qwerty keyboards in the three keyboard conditions. Similarly, the error rate of the dynamic split-key keyboard shows no significant difference between the full qwerty keyboard in the three keyboard conditions. Our hypothesis is valid for the text entry speed and error rate in stylus use.

The results support our hypothesis that the dynamic split-key software keyboard design for a touch screen in finger use offers equal performance to that of a static split-key keyboard. The dynamic split-key keyboard's text entry speed is not significantly different from that of the static split-key keyboard in the three keyboard conditions, although it shows no significant difference from the full qwerty keyboard. The error rate of the dynamic split-key keyboard shows no significant difference from the static split-key keyboard, but it is significantly different from the full qwerty keyboard, specifically in the small and medium keyboard conditions. No significant difference was found among the three keyboard conditions for the large keyboard. Therefore, our hypothesis is validated for the text entry speed and error rate in finger use.

Conclusion

As described in this paper, we proposed a flexible split-key interface for a touch screen software keyboard, which can be adjusted for either stylus use or finger use. Evaluations show that our proposed approach has equal performance to that of the full qwerty keyboard for stylus use and to that of the static split-key keyboard for finger use.

Future studies might be undertaken to design an appropriate pop-up menu. The current interface uses a pie menu. The menu items change from time to time depending on the user's touch area. This dynamism might decrease the text input performance because the user must always confirm the position of the targeting menu item in a dynamically changeable menu.

References

[1] Hwang, S. and Lee, G. Qwerty-like 3x4 keypad layouts for mobile phone. Ext. Abstracts CHI 2005, ACM Press (2005), 1479-1482.

[2] Gong, J. and Tarasewich, P. Alphabetically constrained keypad designs for text entry on mobile devices. Proc. CHI 2005, ACM Press (2005), 211-220.

[3] Green, N., Kruger, J., Faldu, C., and Amant, R.S. A reduced QWERTY keyboard for mobile text entry. Ext. Abstracts CHI 2004, ACM Press (2004), 1429-1432.

[4] Tanaka-Ishii, K., Inutsuka, Y., and Takeichi, M. Entering text with a four-button device. Proc. 19th ICCL, (2002) Vol. 1, 1-7.

[5] MacKenzie, I.S. and Soukoreff, R.W. Text entry for mobile computing: Models and methods, theory and practice. HCI, Vol. 17, 147-198, 2002.

[6] Vogel, D. and Baudisch, P. Shift: A Technique for Operating Pen-Based Interfaces Using Touch. Proc. CHI 2007, ACM Press (2007), 657-666.