SIMKEYS: An Efficient Keypad Configuration for Mobile Communications

Rick W. Ha, Pin-Han Ho, and Xuemin (Sherman) Shen, University of Waterloo

ABSTRACT

Although text messaging services are becoming increasingly popular in today's global wireless market, fundamental design issues still linger with respect to text entry methods on mobile communication devices. Current methods may often be plagued with problems such as poor typing efficiency, stringent physical size limitations, and an unwarranted cognitive processing burden on mobile users. The proposed text entry method for mobile communication devices, called SIMKEYS, balances input efficiency, ergonomics, usability, and cost via a compact 12-button keypad. Pursuing a deterministic and linguistically optimized approach to character disambiguation, SIMKEYS achieves significant improvement in typing performance over existing methods, verified by extensive simulation results. It also consumes negligible amounts of system resources and incurs minimal development costs. Because of its simplicity and efficiency, SIMKEYS opens the door to exciting opportunities in the next stage of development of the wireless Internet.

INTRODUCTION

Since the widespread proliferation of cellular telephone networks and wireless devices in the 1990s, mobile email and short messaging applications continue to experience extraordinary growth across the world. Current statistics indicate that short messaging service (SMS) offered by Global System for Mobile Communications (GSM), the most dominant cellular communications system in the world, generates traffic exceeding 35 billion messages/mo [1]. Similar to the way email applications expedited the Internet's maturity from its infancy stages, mobile short messaging appears to have become the main driving force in fueling the development of wireless Internet access in the near future.

Among the many unique technical challenges faced by communication engineers in implementing ubiquitous Internet access, one subtle obstacle lies in designing a simple and efficient user interface for text entry on mobile devices. These wireless gadgets often do not possess enough physical space to accommodate the complete keyboard configuration available to their full-size computing counterparts. Therefore, a single button on the input keypad may be assigned to more than one character, thus creating plurality in character resolution that requires disambiguation schemes to identify the original character intended by the user. Similar data entry issues exist in other technical applications as well, such as personal digital assistants (PDAs), wearable computers, interactive TV, automotive navigation systems, and smart appliances, and an efficient solution for text entry is urgently needed to foster the next stage of development in these emerging fields.

Besides relying on the physical keypad, several innovative approaches such as touch-screen keypads, character recognition, and virtual keyboards have been actively pursued to support mobile text entry. While each approach is associated with various benefits, their respective limitations make them not applicable to all situations. For example, touch-screen keypads can be loaded with a full keyboard for access via fingers or a stylus, but restrictions on device dimensions often lead to very small on-screen key size that may be difficult to see and select. On the other hand, not all mobile devices can be equipped with specialized hardware and adequate processing power to support character recognition and virtual keyboards. Therefore, a more concerted research effort should be directed to a text entry solution based on physical keypads whose applicability encompasses most mobile devices.

An investigation into ways to enhance performance and usability in character disambiguation via a 12-button telephone keypad culminated in the development of SIMKEYS, a highly efficient keypad configuration for mobile communications that is the primary focus of this article. As it offers unparalleled performance improvement over existing methods, SIMKEYS will undoubt-

 0163-6804/04/$20.00 © 2004 IEEE

edly be instrumental in facilitating future development of mobile computing devices.

The rest of this article is organized as follows. We provide a general description and the design drawbacks of the current mobile text input methods. Specifics of the proposed SIMKEYS method are introduced, while corresponding performance analysis and comparisons are also presented. We summarize the article with some concluding remarks.

MOBILE TEXT INPUT METHODS

The current text messaging input methods via physical keypads can be classified into three categories: full keyboard, deterministic approaches, and predictive algorithms; each with its own distinct advantages and drawbacks.

FULL KEYBOARD

Because of the ubiquitous penetration of the current full keyboard design, commonly referred to as the QWERTY keyboard, in the computing world, designers are prompted to try their utmost to incorporate the same key configuration or variants of it onto mobile devices. However, this notion seems to ignore some fundamental issues that are pertinent to the mobile environment. From a user's perspective, the smaller physical size of mobile devices causes the keys on the keyboard to cluster together, thus drastically diminishing the possibility of applying touch typing techniques for efficiency improvement. Similarly, with such a large concentration of buttons on very limited physical space, the user will likely experience some trouble in searching for and actually typing the keys. As the user usually relies on both thumbs to perform typing, the extra distance spanning different keys makes the full keyboard not conducive to ergonomics and one-handed operation.

On the other hand, mobile device designers will be vexed by the fact that the large area occupied by the keys leaves less leeway in device design. Since newer versions of mobile devices tend to successively decrease in size, the full keyboard will most likely not be able to fit onto these miniature modules. More important, a full keyboard will demand extra circuitry and elicit more battery power from mobile devices that are already energy deficient. This leads to additional development costs that will be transferred onto consumers, who will certainly be dissatisfied. Therefore, regardless of how accustomed the public is to the universal QWERTY keyboard, these adverse factors should certainly justify the adoption of other methods of mobile text entry.

DETERMINISTIC METHODS

In order to reduce the number of keys installed on mobile devices, some ambiguity must be introduced in the character identification process. For deterministic methods, mobile users would enter a distinct combination of keys consecutively to unambiguously represent a certain character. Currently available methods that are based on the 12-key telephone keypad but with different character assignments include *Multitap*, *Less-Tap*, *MessagEase* and *Two-Key*. The first two allow variable keystrokes per character from one to four, while the latter two assume a fixed two keystrokes per character.

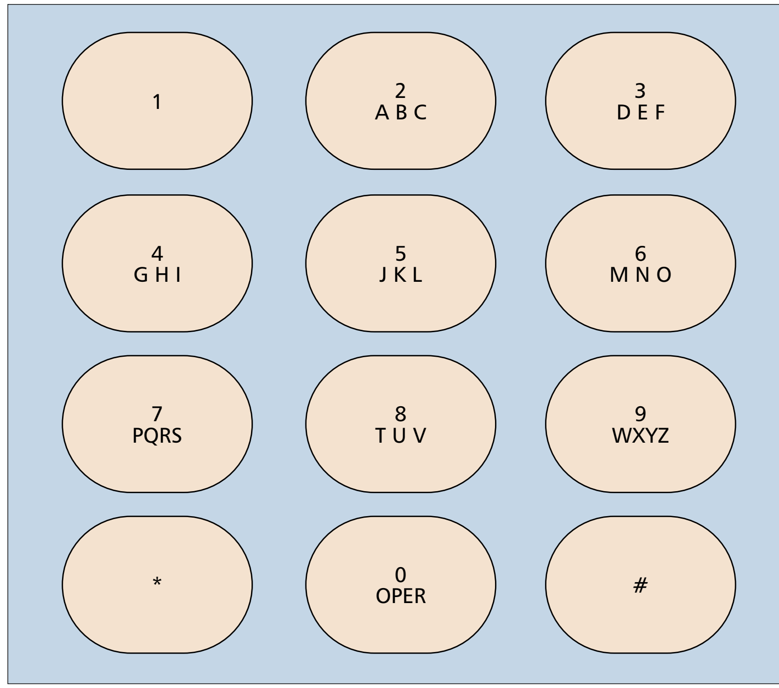

Figure 1. *The standard telephone keypad layout.*

Multitap, which is based on the alphabetic telephone keypad shown in Fig. 1, has practically become the standard text entry method for mobile communications because of its seamless integration with existing telephony applications. Its operations are straightforward: a particular button is pressed *i* times to select the *i*th character located on that button. If two consecutive characters reside on the same button, the user would need to wait for a timeout or press a *timeout-kill* button in between to reset the internal state machine. Although Multitap allows easy memorization of the character assignments and achieves seamless integration with the current telephone system, there exist a number of problems regarding its efficiency and usability. First, some characters that are frequently used require more than one tap, while letters that are seldom used often involve fewer taps during typing. This lettering arrangement thus leads to very high keystrokes per character and subsequently a lower realizable typing speed. Second, both instances of waiting for a full timeout and pressing an explicit button to delimit the characters of the same button result in lower typing efficiency. A Multitap-based scheme called Less-Tap, shown in Fig. 2a, improves typing efficiency by reconfiguring the character layout such that more frequently used characters are given precedent by requiring fewer keystrokes [2]. Nevertheless, the segregation of consecutive characters still requires the use of timeouts or an explicit key that would degrade typing speed.

Figure 2b shows the character layout of MessagEase, which assigns unique key combinations according to the topological location of the letters and symbols on the keypad [3]. While this method improves efficiency by allocating the

1	Space	18.66%	10	H	4.20%	19	P	1.46%
2	E	10.83%	11	D	3.64%	20	W	1.32%
3	T	7.97%	12	L	3.52%	21	B	1.25%
4	A	6.61%	13	F	2.50%	22	V	0.87%
5	O	6.28%	14	C	2.33%	23	K	0.81%
6	N	5.39%	15	M	2.01%	24	X	0.13%
7	R	4.94%	16	U	1.92%	25	J	0.11%
8	I	4.90%	17	G	1.77%	26	Q	0.06%
9	S	4.78%	18	Y	1.70%	27	Z	0.05%

Table 1. *Frequency ranking of letters and space in English.*

most frequently used characters in English with combinations that just require double pressing of the same key, the scattered letter layout makes memorization very time-consuming. A variant of this approach, commonly known as Two-Key, maps the familiar telephone keypad shown in Fig. 1 with a coordinate system where the first keystroke locates the group of letters and the second keystroke identifies the location of the letter within the group. Although this method does not stipulate lettering rearrangements from the alphabetical keypad configuration, it still requires two keystrokes per character, and fails to exploit linguistic characteristics that will substantially improve typing efficiency.

Predictive Algorithms

Compared to full keyboards, deterministic methods are simpler to implement on mobile devices with fewer buttons, but the resultant keypad layouts require character disambiguation methods that often drastically reduce input efficiency. The investigation into other approaches that combine simplicity and efficiency led to the development of predictive algorithms, which originally serve the purpose of facilitating text entry and computer access for individuals with disabilities via a keypad or reduced keyboard [4]. Instead of manually delimiting each character, predictive methods associate input key sequences to distinct words according to linguistic statistics. In particular, predictive methods direct users to enter *groups of characters* per keystroke rather than individual letters and let the built-in algorithms decipher the input key sequences into most probably intended words and phrases. Predictive methods such as *T9* and *LetterWise* purport to achieve performance close to one keystroke per character [5, 6].

Although predictive algorithms generally achieve higher typing efficiency over deterministic approaches, there exist a number of problems that severely hamper their operation. First, predictive methods realize lower keystrokes per character at the expense of higher demands on mobile processing and memory storage. Second, the increase in typing efficiency only comes when proper vocabulary usage and perfect spelling are in effect, and spelling mistakes tend to be difficult to correct without re-entering the entire sequence. Third, implementing predictive algorithms for different languages will require separate linguistic modules, thus increasing development and licensing costs for device manufacturers. Fourth, although fewer keystrokes are required per character via predictive methods, users will inadvertently expend additional mental processing power to manually verify the prediction results after entering the key sequences. Therefore, overall typing efficiency may be degraded in spite of the gains achieved by character prediction.

SIMKEYS Specifications

Given the deficiencies of the current text input methods, the concept of SIMKEYS is conceived in an attempt to incorporate design features such that efficiency, ergonomics, usability, and costs in mobile text messaging can be balanced. Upon analyzing the advantages and drawbacks of various methods, a deterministic approach is pursued by SIMKEYS because of its structural simplicity, which translates into ease of use and low development costs. To enhance typing efficiency, a completely revamped keypad design linguistically optimized for text entry replaces the universal alphabetic telephone layout that is

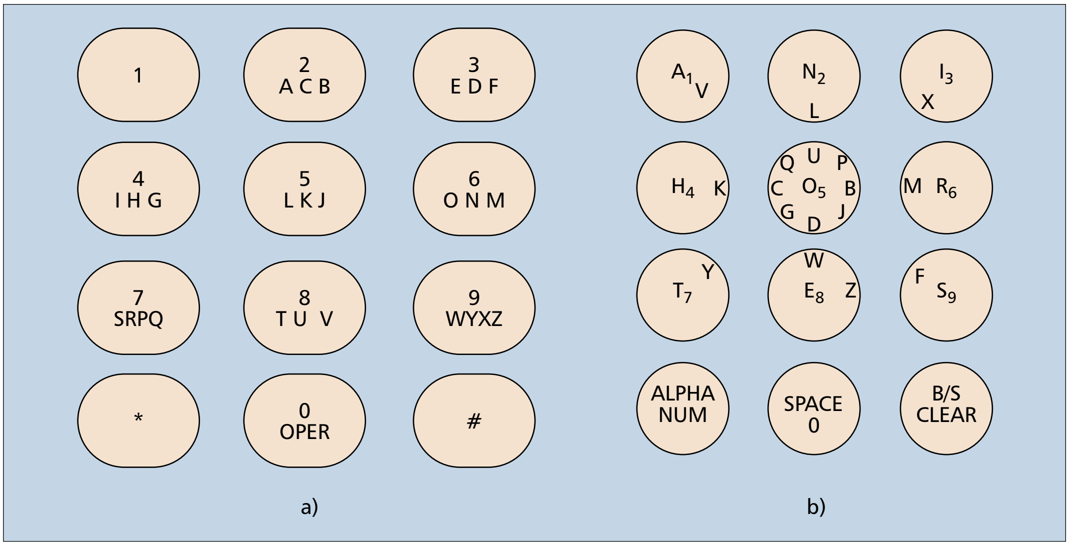

Figure 2. *Keypad layouts: a) Less-Tap; b) MessagEase.*

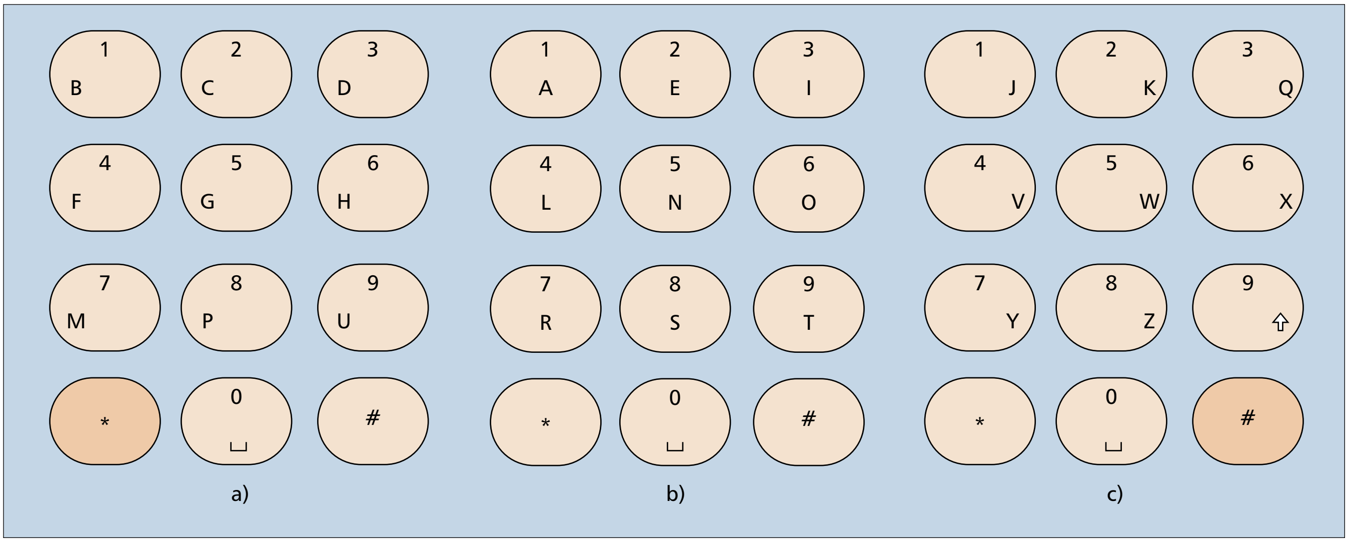

Figure 3. *SIMKEYS character planes: a) star plane; b) center plane; c) pound plane.*

ill equipped for text entry. Furthermore, SIMKEYS is based on a deterministic model that uses neither a timer for letter disambiguation nor a prediction algorithm for letter entry. The novel SIMKEYS design signifies major improvements in typing efficiency while consuming minimal system resources.

Because of the legacy of the alphabetic telephone keypad and its influence on Multitap-based and Two-Key methods, the characters are all concentrated on buttons 2 to 9 while keys 0, 1, *, and # are often assigned auxiliary functions such as shifting, symbols, space, and character disambiguation. SIMKEYS allocates fewer keystrokes to more frequently used characters in order to increase typing efficiency, similar to Less-Tap and MessagEase. However, the style in which keystrokes are assigned is quite different from that of the Multitap method. The first step in assembling the new SIMKEYS keypad layout is to determine the usage frequency of all the letters in the alphabet. Table 1 shows the frequency ranking of letters and space in English [7]. Similar character occurrence patterns have also been found in other major Latin-alphabet-based languages such as French, German, and Spanish [8]. Based on this, the SIMKEYS letter configuration would be arranged according to character frequency rather than character ordering within the alphabet.

Although in MessagEase all the characters require two keystrokes, the nine most frequently used characters are placed on individual number keys such that each letter only requires a double click of its respective button. SIMKEYS goes a step further by dictating only one keystroke for the most frequently used characters, and the character placement is shown in Fig. 3b. In addition, the 0 button is dedicated as the space key since it is often more frequently used than any other character. These frequently used characters (i.e., A-E-I-L-N-O-R-S-T), which only require one keystroke per letter, constitute the *center plane* of SIMKEYS.

In contrast to MessagEase and Two-Key where a coordinate-based system is sought for mapping characters to keys, SIMKEYS follows an approach that is similar to how the shift key functions on a regular keyboard for assigning the rest of the characters onto the 12-key keypad. Much like how the shift key divides the standard keyboard into two planes of upper and lower case characters, SIMKEYS exploits the seldom-used star and pound keys on the telephone keypad to generate additional unique key combinations for accommodating the entire Latin alphabet. Let the remaining characters be separated into two groups, each containsing the second most and least frequently used characters, respectively. To access the first group (i.e., B-C-D-F-G-H-M-P-U), the user would first press the star key and then the corresponding number key. Likewise, the latter group (i.e., J-K-Q-V-W-X-Y-Z) is accessed via the pound key. These two groups of letters are referred to as the *star plane* and *pound plane*, and their arrangements are shown in Fig. 3a and c. Since the pound plane only has eight characters, the extra ninth slot is allocated to the shift function. Upper case letters can be entered by first pressing the key sequence of #-9 to activate the shift function followed by typing the desired letter. This method adequately handles entry of capital letters if they appear infrequently during message composition. Figure 4 presents the final SIMKEYS layout after all three character planes are combined.

As text messages often involve a combination of lower and upper case characters, numbers and symbols, and so on, SIMKEYS proposes a new key called *mode*, shown in Fig. 4, to interchange the character entry modes of the telephone keypad. Four character entry modes are specified: normal (lower case), caps lock (upper case), num lock (numbers), and symbol. By successively pressing the *mode* key, the entry mode of the keypad transfers from normal to caps lock, then to num lock, then to symbol, then back to normal, and so forth. Just like the many mobile messaging applications available today, symbol entry can be performed via a list or a multipage array format, both of which are expandable to include more symbols and icons in the future. By embracing the mode button format, SIMKEYS

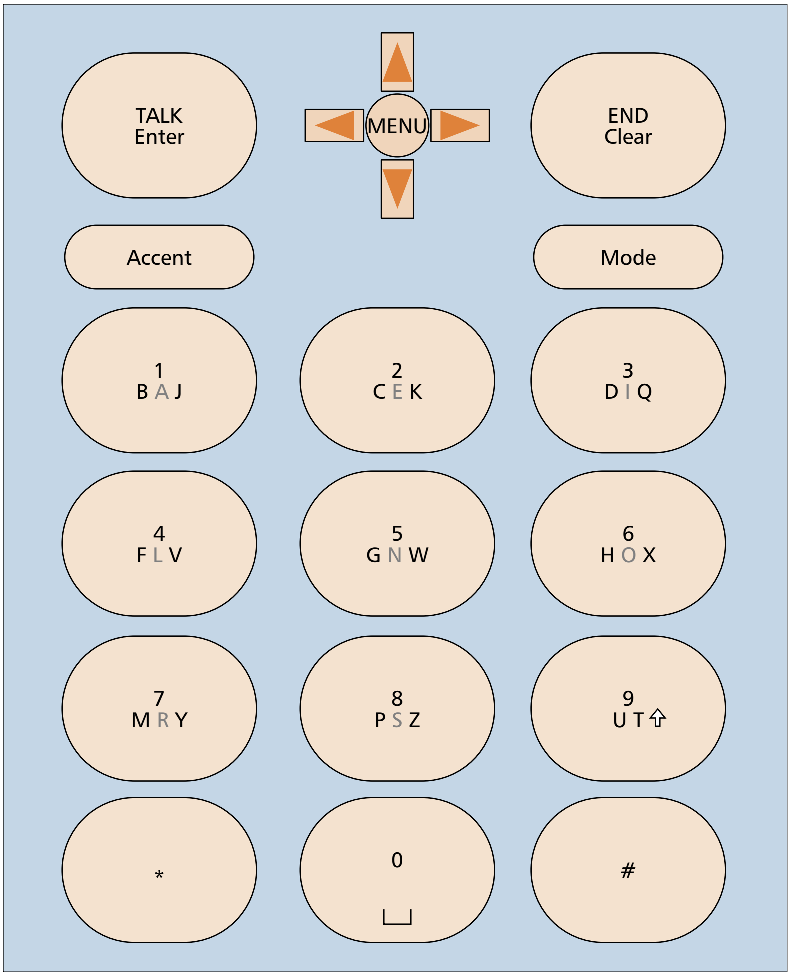

Figure 4. *SIMKEYS keypad layout with system function buttons.*

reduces the need for separate functional keys and maximizes key reuse for the telephone keypad, both of which are beneficial to conserving precious system resources in the mobile operating environment. On the other hand, frequently used functions such as enter and backspace can be assimilated into the available keys on a mobile phone, such as talk and end.

In addition to the regular 26-letter Latin alphabet used in English, many other Latin-alphabet-based languages supplement the normal alphabet set with accented and special characters. Since the three character planes in SIMKEYS together only have enough slots for the 27 elementary letters, any extra character must be allocated in one of the following two ways. First, they can be classified as universal symbols so that they can be incorporated in all language versions of mobile messaging services. However, this approach may be as a nuisance since the user needs to invoke the symbols entry interface each time such characters are encountered. Another approach is to let the text entry program automatically generate the appropriate notation with reference to the words typed. While this approach may be clever, it suffers from the same deficiencies common to predictive algorithms such as increased memory demands and poor cross-lingual adaptability. The third method suggests the inclusion of a new key called *accent*, also shown in Fig. 4, such that accented and special characters are associated with the corresponding regular letter in the alphabet. For instance, to enter accented versions of the letter e in French (i.e., é, è, ê, and ë), the user would first press the *2* button and then select the desired character by successively pressing the Accent button. This method entails customizing the default SIMKEYS interface to specific languages, which may incur additional costs, albeit relatively minor, in development and logistics. In summary, mobile device manufacturers and wireless carriers should be able to provide downloading and upgrading of multilingual SIMKEYS input methods with little technical difficulty.

In the SMS universe there are specialized terms, commonly referred to as *lingo*, that are sometimes undecipherable by the untrained eye. These terms are often abbreviations of standard text (e.g., tmr for tomorrow), borrowed words from foreign languages, or outright concoctions out of users' imaginations. Because SIMKEYS is not dictionary-based, users will encounter very little difficulty in entering lingo via SIMKEYS. Also, even though some of the letters that appear frequently in lingo text may not be commonly used characters, the requirement in typing is at most two keystrokes per character. Therefore, SIMKEYS is still able to offer increased efficiency and adaptability in the ever evolving realm of lingo usage.

Performance Analysis

In order to evaluate the effectiveness of SIMKEYS in comparison to other methods, a set of performance metrics is specified and computed. Since most of the other input methods do not stipulate standardized approaches in handling case switching and entry of numerals and symbols, the performance analysis below will be only on lower case alphabetical character entries.

Keystroke per Character

In general, the most fundamental performance metric for any input method is its keystroke per character (KSPC) figure, which indicates the average number of key presses to type a character. The formula to calculate KSPC is

$$KSPC = \sum_{i=1}^{26} P_i \times n_i, \qquad (1)$$

where P_i is the probability of occurrence for the ith character of the alphabet, and n_i is the number of keystrokes to type the ith character. The KSPC for the various entry methods is evaluated using linguistic statistics as listed in Table 1 for every letter excluding space, and the results are presented in Table 2. Note that the KPSC figures for T9 and LetterWise are directly drawn from [9] based on similar linguistic statistics. From Table 2, both Two-Key and MessagEase require two keystrokes for every letter of the alphabet. Predictive methods such as T9 and LetterWise fare quite well as their KSPC approaches 1, but both are based on the generous assumption that only dictionary-based English words are entered. Among the deterministic methods, it is clear that SIMKEYS prevails over Multitap and Less-Tap in KSPC evaluation. Compared to the ideal 1 KSPC figure, Multitap and Less-Tap require 150 and 55 percent in keystroke overhead when typing English, respectively. SIMKEYS only incurs around 30 percent overhead, which is a near 40 percent improvement in efficiency over Multitap.

Words per Minute

Another effective performance metric for measuring the efficiency of an input method is words per minute (WPM), which concerns the projected typing speed of users. The major factors in determining WPM are keypad dimensions, linguistic correlations, finger movement quickness, and cognitive delays. The general formula in calculating WPM is given by

$$WPM = \frac{60}{CPW \times TPC}, \tag{2}$$

where CPW, short for characters per word, indicates the average number of characters that constitutes each word in the English language, which is roughly equal to 5 characters/word. TPC stands for time per character, which represents the average time required to type a character. It can be calculated via an analytical model that considers the combined effects of finger movements, cognitive processing, and linguistic statistics on typing speed. In short, the model can be represented by the following formula:

$$TPC = \sum_{i=1}^{27}\sum_{j=1}^{27} P_{ij} \times CT_{ij}, \tag{3}$$

where P_{ij} denotes the probability of letter-pair (digraph) i-j consecutively appearing in text, and CT_{ij} is the character entry time for letter j given i has appeared. P_{ij} values are conveniently drawn from Mayzner and Tresselt's table, which is essentially a 27 × 27 alphabet matrix that documents the relative frequencies of digraphs from a large number of vocabulary samples in a variety of English media sources [7]. The derivation for CT_{ij} is more complex since it involves finger movements and cognitive processing, both of which rely heavily on physical dimensions of the mobile device as well as empirical data obtained from previous usability studies. In the current WPM analysis, two methods of modeling CT_{ij} are studied and compared. The first one, described in [9], only considers finger movement as the sole determinant of CT_{ij}, which in essence assumes that the users are expert typists with little mental hesitation when entering consecutive characters on telephone keypads. The second model, introduced in [10], incorporates both finger movement and cognitive processing overhead in computing CT_{ij} such that the behaviors of nonexpert and average mobile messaging users can be portrayed. In essence, the total input time is the sum of the time used for finger movement and any associated cognitive delays, in which the latter values are determined through empirical evidence collected from extensive user trials as noted in [10]. As different text entry methods are associated with distinct finger movement and cognitive delay patterns, the exact model formulation will vary in each case.

Two-Key/MessagEase	T9	LetterWise	Multitap	Less-Tap	SIMKEYS
~2	1.0072	1.15	2.15	1.55	1.30

Table 2. *KSPC for a number of input methods.*

To quantify the entry time elapsed through finger movements and the actual keystroke on a telephone keypad, a simple mathematical expression called Fitts' Law can be applied [11], which takes the form

$$MT = a + b\log_2(A/W + 1), \tag{4}$$

where a and b are system coefficients that depend on the method of entry, A is the length or amplitude of movement, and W is the target size. For every input method in the current study, it is assumed that the user relies on only a single thumb to perform text entry. Subsequently, coefficients a and b were experimentally verified to be 176 ms/b and 64 ms/b, respectively, for one-thumb typing mode [29]. On the other hand, parameters A and W both depend on the physical dimensions of the 12-button telephone keypad tested, which has button sizes of 6 mm × 10 mm and interkey distances of 9–11 mm.

Table 3 presents the projected WPM values for different input methods with and without considering user cognitive delays. It can be seen that SIMKEYS outperforms all other deterministic input methods by a significant margin with or without considering cognitive delays. In particular, SIMKEYS offers 47 and 31 percent higher improvement in efficiency over Multitap in both models, respectively. Despite the seemingly considerable difference in WPM between SIMKEYS and predictive methods, the actual performance difference is much smaller in practice since predictive methods were supplied with the generous assumption of perfect disambiguation and word prediction. Because of the sub-100 percent prediction accuracy of predictive algorithms in reality, the user will habitually verify the prediction results after entry completion of each word, thus incurring additional cognitive processing overhead.

Conclusions

SIMKEYS, an efficient keypad configuration that offers a fresh perspective in implementing text entry on mobile devices, has been proposed. Rather than relying on sophisticated user interface approaches, SIMKEYS is based on a deterministic approach that exploits linguistic statistics to achieve simplicity and performance improvement using a modified 12-button telephone keypad configuration while incurring minimal system resources in the mobile computing environment. The meticulous SIMKEYS keypad layout is optimized for most languages based on the Latin alphabet, and it seeks to be a global standard for ambiguous text input. Extensive analytical results have proven that SIMKEYS prevails over existing methods by a considerable margin in several performance metrics. The next steps in SIMKEYS development include usability trials and industry feedback to further refine the novel keypad design for better memorization and a

Because of its simplicity and efficiency, SIMKEYS opens the door for exciting opportunities in product design for mobile communications, and it will certainly help to pave the way towards the next stage of wireless Internet development.

Method	WPM (without cognitive delay)	Difference	WPM (with cognitive delay)	Difference
Predictive	40.65	24.99%	9.99	7.57%
SIMKEYS	30.49	—	9.23	—
Less–Tap	25.13	–21.36%	7.77	–18.90%
MessagEase	25.00	–21.97%	8.66	–6.65%
Two–Key	22.43	–35.97%	8.33	–10.89%
Multitap	20.73	–47.12%	7.03	–31.38%

Table 3. *Projected WPM values for different input methods with or without cognitive delay.*

quicker learning process.

Fully embracing the SIMKEYS concept may take time and genuine conviction from both the wireless industry and the user population because its implementation entails a radical key reconfiguration of the familiar alphabetical telephone keypad that could impact existing telephone dialing practices such as vanity numbers (e.g., 1-800-SIMKEYS). With clever user interface design, however, the SIMKEYS configuration can coexist with the current telephone keypad layout such that users can switch between operation modes with ease. It is expected the general public will soon discover that the benefits of using SIMKEYS for text entry will far outweigh its few disadvantages in the long run. Also, the use of SIMKEYS does not preclude its integration with predictive methods on mobile devices, which together will offer users additional performance enhancements in text entry. Because of its simplicity and efficiency, SIMKEYS opens the door to exciting opportunities in product design for mobile communications, and will certainly help pave the way toward the next stage of wireless Internet development.

References

[1] GSM World statistics; http://www.gsmworld.com /technology/sms/index.shtml

[2] A. Pavlovych and W. Stuerzlinger, "Less-Tap: A Fast and Easy-to-Learn Text Input Technique for Phones," *Proc. Conf. Graphics Interface*, Halifax, Nova Scotia, Canada, June 2003.

[3] S. B. Nesbat, "A Fast, Full-Text Entry System for Small Electronic Devices," White paper, Exideas Inc., 2001.

[4] G. W. Lesher, B. J. Mouton, and D. J. Higginbotham, "Optimal Character Arrangements for Ambiguous Keyboards," *IEEE Trans. Rehab. Eng.*, vol. 6, no. 4, 1998, pp. 415–23.

[5] D. L. Grover, M. T. King, and C. A. Kushler, "Reduced Keyboard Disambiguating Computer," U.S. Patent 5,818,437, to Tegic Communications, Inc., 1998.

[6] I. S. MacKenzie *et al.*, "LetterWise: Prefix-Based Disambiguation for Mobile Text Input," *Proc. ACM Symp. User Interface Software and Tech.*, 2001, pp. 111–20.

[7] M. S. Mayzner and M. E. Tresselt, "Tables of Single-Letter and Digram Frequency Counts for Various Word-Length and Letter-Position Combinations," *Psychonomic Monograph Supplements*, 1965, pp. 13–32.

[8] F. Pratt, *Secret and Urgent: The Story of Codes and Ciphers*, Blue Ribbon Books, New York, 1939, pp. 253–54.

[9] M. Silfverberg, I. S. MacKenzie. and P. Korhonen, "Predicting Text Entry Speeds on Mobile Phones," *Proc. ACM Conf. Human Factors in Comp. Sys.*, New York, 2000, pp. 9–16.

[10] A. Pavlovych and W. Stuerzlinger, "Model for Non-Expert Text Entry Speed on 12-Button Phone Keypads," Tech. rep., York Univ., Toronto, Canada, 2003.

[11] P. M. Fitts, "The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement," *J. Exp. Psych. 47*, 1954, pp. 381–91.

Biographies

RICK W. HA (rwkha@bbcr.uwaterloo.ca) received his B.A.Sc. in computer engineering and M.A.Sc. in electrical and computer engineering in 2000 and 2002, respectively, from the University of Waterloo, Canada, where he is currently pursuing his Ph.D. degree. His research interests include cross-layer design of wireless ad hoc and sensor networks.

PIN-HAN HO (pinhan@bbcr.uwaterloo.ca) received his B.Sc. and M.Sc. degrees from the Electrical and Computer Engineering Department of National Taiwan University in 1993 and 1995. He started his Ph.D. study in 2000 at Queen's University, Kingston, Canada, focusing on optical communications systems, survivable networking, and QoS routing problems. He finished his Ph.D. in 2002, and joined the Department of Electrical and Computer Engineering, University of Waterloo, as an assistant professor. He is the first author of more than 60 refereed technical papers and book chapters, and co-author of a book on optical networking and survivability. He is the recipient of the Best Paper Award and Outstanding Paper Award from SPECTS '02 and HPSR '02, respectively.

XUEMIN (SHERMAN) SHEN [SM] (xshen@bbcr.uwaterloo.ca) received his B.Sc. (1982) degree from Dalian Maritime University, China, and his M.Sc. (1987) and Ph.D. degrees (1990) from Rutgers University, New Jersey, all in electrical engineering. From September 1990 to September 1993 he was first with Howard University, Washington DC, and then the University of Alberta, Edmonton, Canada. Since October 1993 he has been with the Department of Electrical and Computer Engineering, University of Waterloo, where he is a professor. His research focuses on mobility and resource management in interconnected wireless/wireline networks, UWB wireless communications, ad hoc networks, and sensor networks. He is a co-author of two books, and has publications in communications networks, control and filtering. He has served as Technical Co-Chair for IEEE GLOBECOM '03 Symposium on Next Generation Networks and Internet and ISPAN '04; as Editor for *IEEE Transactions on Wireless Communications*, *IEEE Transactions on. Vehicular Technology*, and three other journals; and Guest Editor for *IEEE Journal on Selected Areas in Communications*, Special Issue on Ultra Wideband Wireless Communications; *IEEE Communications Magazine*, Special Issue on China Wireless Communications: Technology vs. Markets, and *IEEE Wireless Communications*, Special Issue on 4G Mobile Communication: Toward Open Wireless Architecture. He received the Premier's Research Excellence Award (PREA) from the Province of Ontario for demonstrated excellence of scientific and academic contributions, and the Distinguished Performance Award from the Faculty of Engineering, University of Waterloo, for outstanding contributions in teaching, scholarship, and service. He is a registered Professional Engineer of Ontario, Canada.