Applying Mobile Device Soft Keyboards to Collaborative Multitouch Tabletop Displays: Design and Evaluation

Sungahn Ko,[1] *KyungTae Kim,*[2] *Tejas Kulkarni,*[3] *Niklas Elmqvist*[1]

[1]Purdue University
West Lafayette, IN, USA
{ko, elm}@purdue.edu

[2]LG Electronics
Seoul, South Korea
kyungt.kim@lge.com

[3]Qualcomm
San Diego, CA, USA
tejask@qualcomm.com

ABSTRACT

We present an evaluation of text entry methods for tabletop displays given small display space allocations, an increasingly important design constraint as tabletops become collaborative platforms. Small space is already a requirement of mobile text entry methods, and these can often be easily ported to tabletop settings. The purpose of this work is to determine whether these mobile text entry methods are equally useful for tabletop displays, or whether there are unique aspects of text entry on large, horizontal surfaces that influence design. Our evaluation consists of two studies designed to elicit differences between the mobile and tabletop domains. Results show that standard soft keyboards perform best, even at small space allocations. Furthermore, occlusion-reduction methods like Shift do not yield significant improvements to text entry; we speculate that this is due to the low ratio of resolution per surface units (i.e., DPI) for current tabletops.

ACM Classification: H.5.2 [Information Interfaces and Presentation]: User Interfaces—*Interaction styles*; I.3.6 [Computer Graphics]: Methodology and Techniques—*Interaction techniques*

Keywords: Text entry, computer-supported collaborative work (CSCW), tabletop displays, collaboration.

INTRODUCTION

Digital tabletops are examples of large displays, and so maintaining a limited screen footprint is not a typical design constraint for interaction techniques developed for a tabletop in the same way they are for small displays, such as on mobile devices. However, as tabletops are increasingly being used for co-located collaboration (e.g., [2, 28]), screen real estate remains a scarce commodity when additional participants gather around the table to work together [13]. In other words, maintaining a small display footprint is becoming an important additional factor for interaction design of tasks involving multiple collaborators on these tabletops.

Text entry is one such task that is common to many co-located collaborative applications; consider examples such as collaborative tagging [8], collaborative visualization [16], or even writing small snippets of text, such as for reports, status updates, or labels. Because of its ubiquity in computing, there has been considerable research on general text entry—see for example [4, 5]. However, text entry using physical keyboards is impractical on the kind of collaborative tabletop platforms discussed here—there is typically no good place to park the keyboard, and each collaborator will often need their own keyboard. Instead, designers of collaborative software often turn to soft keyboards for tabletop text entry.

A *soft keyboard* [21, 23] is a virtual keyboard that is displayed on the screen instead of having a physical form. This allows us to overcome both the parking and availability problems discussed above—a soft keyboard can be hidden from the screen when it is not in use [13], and there is no real limit to the number of keyboards that can be displayed at a time. Keyboards are generally evaluated on their efficiency—the typing speed achievable by an experienced user—and their ease of learning [23]—the time and effort required to become proficient with the keyboard. These factors are often conflicting. However, just like for mobile text entry, additional factors come into play for text entry on tabletop displays:

C1 **Space:** The screen footprint of the keyboard's visual representation (i.e., the amount of pixels it consumes);
C2 **Occlusion:** The impact of finger occlusion (problematic for soft keyboards due to the lack of tactile feedback); and
C3 **Robustness:** The method's tolerance against input/output calibration errors or low input resolution.

Like physical keyboards, soft keyboards have also been exhaustively researched over the years; see for example [21, 23, 31]. However, these techniques are all designed for single-user settings, and it is not always clear how well they will work for tabletop displays, where additional constraints such as rotatability, direct touch interaction, mobility, and simultaneous interaction may have an impact [13] . Even for mobile text entry methods (e.g., [24, 26]), which share the above design criteria (particularly the limited screen footprint), there is little work on how well existing methods translate to the unique properties of the tabletop platform.

We present an evaluation of text entry techniques on multi-touch tabletop displays under small screen space allocations.

Conference ITS'11, November 13-16, Kobe, Japan.

Our evaluation was designed to elicit differences between the mobile and tabletop domains, and thus studies the performance of a representative sample of mobile text entry techniques adapted to the tabletop. We also include a novel pinpointing technique designed specifically for multitouch text entry for small display footprints. We perform two studies: an initial study, where we establish methods and parameters, and a follow-up study, where we measure text entry performance given these parameters.

RELATED WORK

Text entry is ubiquitous in interactive computing, and before computers there were typewriters; see for example the work by Dvorak et al. [5]. In this paper, we study text entry for collaborative tabletop displays, which gives rise to two main design constraints [13]:

- **Placement:** Tabletops do not provide a natural place for a physical keyboard so that it does not hide the display; and
- **Multiple keyboards:** Collaborators working together on a tabletop often need their own keyboard.

As already discussed, designers and researchers typically turn to *soft keyboards* [21, 23]—keyboards that do not have a physical form, but rather are drawn using interactive graphics on the computer screen—to avoid these problems. In this section, we will examine the state of the art on soft keyboards, including performance, layout, and size, as well as their use on both mobile devices, such as PDAs and smart phones, and on digital tabletops, like in our work.

General Soft Keyboards

Soft keyboards trivially fulfill the above two constraints: because it is virtual, the keyboard can be hidden when not in use, and any number of separate keyboards can be created. However, because soft keyboards only exist on the screen and not in the physical world, users must utilize the input devices of the computer to control them. Most systems where soft keyboards are used have touch-based or pen-based input devices, easing interaction, but the tactile feedback of touching real, physical keys is still missing. This also means that the occlusion of the user's finger on the key to type is exacerbated [15]. While recent work has explored inflatable buttons to provide tactile feedback even on touch displays [11], this is a problem with no easy and general solution.

Given these problems, it is clearly relevant to study the performance of soft keyboards in comparison to physical keyboards. However, the traditional longitudinal evaluation method designed to capture user performance after the initial learning curve of adopting a new text entry method is costly and time-consuming [41]. Alternative methods use predictive models to avoid these costs [13]; examples include work by Soukoreff and MacKenzie [21, 33], where Fitts' law is used to predict the time to tap a key from any previous key. This enables measuring theoretical performance using familiar metrics such as words per minute (WPM).

A large number of virtual keyboard techniques have been proposed—Zhai et al. [42] gives an excellent summary of these techniques and their performance data. However, to our knowledge, there exists no study investigating text entry performance on direct-touch tabletops using soft keyboards.

Layout and Size for Soft Keyboards

Different layouts have different effect on performance—for example, the Dvorak keyboard layout is optimized according to character frequency in the English language [5]. Beyond character frequency, researchers have also proposed alphabetical layouts and layouts based on human physical models [21, 23, 39]. However, we regard layout to be outside of the scope of our work—while it may arguably [25] have an impact on optimizing individual text entry techniques, we simply choose the default (and thus supposedly optimal) layouts for each technique in this work.

In contrast, there are controversial arguments for the relation between performance and size of soft keyboards on mobile devices. MacKenzie et al. [22] insist that the number of errors on smaller soft keyboards increases but found almost no difference in text-entry speed between large and small keyboards; this is backed by Sears et al. [32]. There clearly must exist some cut-off size limit where keyboards no longer work well, but no research has so far focused on finding this limit. Furthermore, the above studies were conducted on mobile devices, and there exists no data for tabletops.

Mobile Text Entry

Mobile devices like cell phones and PDAs pose particular challenges for text entry due to their limited screen size. With the advent of smart phones and mobile applications for web browsing and social computing, text entry has become one of the most common tasks performed on mobile devices. Therefore, mobile computing has been pushing much of the innovation in text entry using soft keyboards in the last several years. While some recent mobile devices are based on touch screen interaction, like the Apple iPhone, many mobile devices support input using a stylus.

Pen-based input clearly lends itself to text entry using hand writing, but hand writing is computationally difficult to recognize without errors. A better approach is to have users input text using stroke gestures that are specifically designed to be recognized and lack the complexities of normal characters. Unistrokes [7] is an example of this idea, and uses single strokes ("unistrokes") to type letters. The Graffiti system employed in older PalmOS devices uses a similar approach [20]. While recognition of these strokes can be fast, and typing can also be quick and efficient once mastered, stroke-based alphabets require additional learning beyond standard typing.

Venolia and Neiberg [35] proposed a text entry method based on selecting letters in a menu. Their idea, T-Cube, uses hierarchical marking menus (cascading pie menus) to let users select characters. The method consumes very little screen space, and when mastered, an expert user can type quickly using memorized menu gestures.

Quikwriting, designed by Perlin [26], uses a radial layout and enables typing using continuous stylus movement with no need for lifting. The letters are arranged by frequency in zones around the center of the keyboard, and are accessed by a gesture specifying the location of the character in each zone. In contrast, Cirrin (Circular Input) [24] uses a similar radial layout but without the zones and gestures, making the technique easier to use. Furthermore, the technique enables users to type several characters as a single, smooth stroke, a feature that can be fully utilized by optimizing the order of letters on the layout based on their co-occurrence frequency.

Tabletop Text Entry

Interactive tabletop displays are becoming increasingly used as collaborative platforms [2, 28]. Despite this, there exists very little work on the ubiquitous text entry task for tabletops [13, 29, 38]. Many of the above techniques can easily be adapted to the tabletop domain, but may need redesigning to accommodate the platform; for example, using finger ink or Graffiti-style input is not practical due to the impreciseness of fingers for drawing [29]. Hinrichs et al. [13] provide a set of criteria for evaluating text entry methods specifically for tabletops, including space requirements, rotatability, direct-touch interaction, mobility, and simultaneous input.

Some existing work alleviates the keyboard placement problem by integrating physical keyboards into the display environment. Hartmann et al. [12] use multiple wireless keyboards, one per user, that are tracked by the tabletop display and that can be used for various text entry tasks in various configurations. SLAP keyboards [37] are silicone-based physical keyboards that are also tracked by the tabletop, and which can be dynamically relabeled with keys to allow for text entry or other tasks. However, both of these techniques require special technologies that may not be available for general tabletop applications. The keyboards also consume display space that may be unexpendable for some tasks.

Beyond these physical approaches, few text entry methods have been developed specifically for tabletops. Schmidt et al. [30] describe a column typing approach, but give only preliminary results for a small-scale study. Ryall et al. [29] propose physical devices for input, but sidestep the issue by stating that tabletops are unsuitable for text entry. Wigdor et al. [38] observed a single tabletop user in ordinary use and concluded that a standard soft keyboard was adequate. BubbleType [14] was designed for a walk-up-and-use scenario on a tabletop installed in a public space. Thus, the main focus of the BubbleType system was providing an intuitive, aesthetic, and self-explanatory interface, and not primarily on providing an efficient and space-delimited text entry method.

Finally, very recent work on tabletop text entry by Findlater et al. [6] studied expert typists performing text input on touch surfaces under different conditions, in particular in settings with *no* visual keyboard and *no* visual feedback. Their results indicate high consistency for key touches, suggesting that fully eyes-free typing may be possible on tabletops.

EXPLORING TABLETOP TEXT ENTRY

Our goal with this work is to target co-located synchronous collaborative work on interactive tabletop displays supporting multiple concurrent points of contact [10] (so-called *multitouch* tabletops). These assumptions are grounded in the multi-user collaborative affordance that tabletops provide [28]. Furthermore, multitouch tabletops are now becoming readily available: commercial ones such as DiamondTouch [3] and Microsoft Surface can be easily acquired on the market, and some devices can even be constructed with some technical skill and a limited budget [10].

We now consider the constraints this setting imposes on the text entry task for a tabletop platform. SLAP keyboards [37] and Hartmann's integrated physical keyboards [12] notwithstanding, it is clear that using a physical keyboard is impractical due to the lack of a natural space to place the keyboard, as well as the need for multiple keyboards for multi-user collaboration [13]. Instead, we turn our attention to soft keyboards [21, 23] that we can easily hide when not in use, and that can be instantiated multiple times, one for each user.

Hinrichs et al. [13] has already proposed a taxonomy of both physical and virtual keyboards for tabletops, and their evaluation criteria serve as excellent starting points for our design. However, given our explicit emphasis on supporting multiple users while minimizing interference [9], our design constraints may be slightly different. Below we replicate the constraints from the introduction (still secondary to the efficiency and learnability constraints of all text entry methods):

C1 **Minimize space:** Multiple concurrent users means that the text input method should consume a minimum of space on the collaborative platform. The smaller the keyboard, the less impact it will have on the shared surface and the smaller the risk of interference [9] will become.

C2 **Minimize occlusion:** Soft keyboards lack tactile feedback, so the user's fingers may occlude keys they are about to press (the "fat finger" problem), or the user's hands may cause ghost input [15]. The text entry method should thus minimize this occlusion effect, while still taking advantage of the direct touch supported by a multitouch tabletop [13].

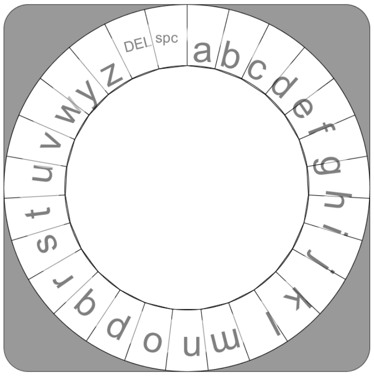
C3 **Maximize robustness:** As the size of touch target decreases, input resolution or minute calibration errors will have increasing impact on the accuracy of the keyboard (particularly on non-commercial or low-cost tabletops). We want to minimize the impact of this error effect.

Our research question in this work is whether text entry is different on multitouch tabletops compared to mobile text entry, which share virtually all of the above design constraints. In other words, we want to determine whether text entry methods designed for mobile devices can be directly ported to a tabletop and perform just as well on the new platform, or whether there are unique aspects of the tabletop display domain that require special consideration.

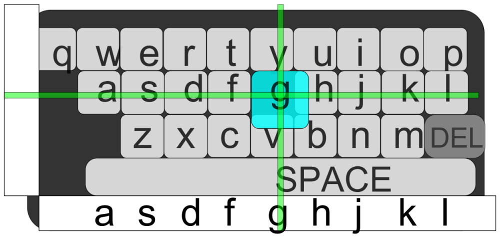
To answer this question, we conducted two user studies investigating human performance for text entry tasks [19, 34] in small display space using four different text entry methods: soft keyboard, soft keyboard with the Shift [36] occlusion-reduction method, radial keyboard [24], and a novel pinpointing text entry method. The initial user study was designed to identify suitable parameters and comparisons; the second follow-up study performed an in-depth evaluation of these conditions.

Beyond the above constraints, we also want to achieve high efficiency [41] while maintaining ease of learning [23] (often conflicting constraints). Therefore, we limit our evaluation to consider only one-key one-character methods that require a minimum of training, thus disregarding chord [27] or keypad-based [18] methods. We also consider keyboard layout [25] to be orthogonal to the research question addressed

(a) Standard soft keyboard.



(b) Radial keyboard.



(c) Pinpoint keyboard.

Figure 1: Keyboard types used in the initial experiment.

in this paper. Optimizing the layout of a particular keyboard will benefit most soft keyboards equivalently, and we therefore choose not to address this issue in our evaluation at all.

This reasoning can be applied to dictionary support that aids typing by suggesting word completions. We therefore disregard this issue. We also do not consider advanced vocabulary text entry methods based on shorthand gestures such as SHARK[2]/ShapeWriter [17, 40], Swype,[1] and BlindType.[2]

USER STUDY

The purpose of our initial user study was to identify a suitable size for the keyboards as well as to make a rough selection of techniques to compare in our in-depth follow-up study.

Apparatus

We used an 1.4m × 0.9m (81") FTIR [10] multitouch tabletop display. The display was set to 1280 × 800 resolution and was powered by a computer running Microsoft Windows.

Participants

Fifteen paid adult volunteers (11 males, 4 females) participated in the experiment. Ages ranged from 21 to 29 (average 24.9, median 25) years, had normal or corrected-to-normal vision, and were experienced, although not professional, typists. Two were left-handed, all others right-handed.

Tasks

We included two tasks with different error correction conditions [1]: None (T1), where users were not allowed to correct errors, and Forced (T2), where they were forced to correct them. In both tasks, users were provided with a soft keyboard and a *Presented Text* (P) [34]. The *Input Stream* (IS) [34] appeared directly below P.

The phrases in our experiment were randomly selected from the standard phrase set for text entry tasks proposed by MacKenzie and Soukorreff [19]. Furthermore, as is standard in text entry evaluation, we removed capitalization and punctuation in each phrase.

Experimental Conditions

We included two factors: keyboard type, and size allocation.

Keyboard Type This factor described the types of soft keyboard that the participants used for performing the text entry. Below we describe the three basic types we included as being representative of the main categories of soft keyboard:

- **Standard soft keyboard (SK):** This was a standard virtual keyboard similar to a physical keyboard with a QWERTY layout. Typing a key amounted to simply touching the relevant key on the graphical representation of the keyboard. Figure 1(a) shows our implementation.
- **Radial keyboard (RK):** This keyboard type was influenced by radial keyboard designs, such as Quikwriting [26] and Cirrin [24], and have not yet been applied to tabletops. Because these keyboards are designed for smooth strokes across a touch surface, they are good candidates for tabletops as well. Figure 1(b) shows our implementation.
  Key layout is a potentially contentious issue on radial keyboards, but we sidestep this issue by simply using alphabetical ordering (the default as indicated by Mankoff and Abowd [24]). This simplifies learning for novice users and removes bias towards special character combinations, which is in line with the general scope of this work.
- **Pinpoint keyboard (PK):** Inspired by some computer game text entry systems, we included a novel text entry method for tabletops called *pinpoint typing*. Instead of tapping keys directly, users input letters by pinpointing the row and column of the key to press using a two-point multitouch gesture, essentially decoupling the horizontal and vertical position of the touch points. This is particularly beneficial for very small size allocations (C1) because it removes occlusion and the fat finger problem entirely (C2), and the visual feedback is robust against minute calibration or resolution problems common in current tabletops (C3). Figure 1(c) shows our implementation.

This list, while hardly exhaustive given the wealth of research on text entry in general, is representative of the largely unexplored design space for tabletops [13]. One potential candidate that we debated including was the BubbleType

---

[1] `http://www.swypeinc.com/`

[2] `http://www.blindtype.com/`

technique proposed by Hinrichs et al. [14]. However, we decided that BubbleTyping mainly concerns graphical layout of keys on screen, and less on the actual interaction technique used to select keys. Therefore, our standard soft keyboard adequately represents the BubbleType technique.

Size Allocation The main hypothesis for this work is that size is an important aspect of soft keyboard design for co-located collaborative work. Therefore, the size allocated to each keyboard type was naturally an important parameter of our experiment. Because the three keyboard types above all use space in different ways, we chose to model this factor as an area as opposed to actual dimensions. For soft and pinpoint keyboards, the keyboard area was computed simply as width × height, whereas for the radial keyboard it was computed as $\pi \cdot r^2$ (where r was the outer radius of the keyboard).

We included three different levels of size allocation in the experiment: small (58 cm^2), medium (175 cm^2), and large (525 cm^2). These values were derived from pilot testing, and for a standard keyboard amounted to 13 × 4.4 cm (small), 23 × 7.6 cm (medium), and 35 × 15 cm (large). While these sizes may not appear very small in comparison to mobile text entry methods, we should note that the screen resolution per physical space (i.e., the DPI) of a tabletop is much coarser than a typical mobile display; in comparison, the smallest keyboard in our experiment was 130 × 50 pixels, whereas an Apple iPhone has a 480 × 320 display and often uses a 240 × 320 pixel soft keyboard (larger in horizontal mode).

It should be noted that this area measurement is not perfectly fair to the radial keyboard, which only uses a ring and not the whole area of the circle. On the other hand, the inner area inside the key ring is reserved for finger strokes for the radial keyboard, and cannot be used for any other purpose—not even showing the visual representation under the keyboard, as this might interfere with text entry. Therefore, we think our choice of size measurement is appropriate.

Design

We used a full factorial within-participant design:

	15	Participants
×	3	**Keyboard Types** *K* (*Standard*, *Pinpoint*, *Radial*)
×	3	**Size Allocations** *A* (*Small*, *Medium*, *Large*)
×	2	**Tasks** (*None* (T1), *Forced* (T2))
×	1	repetition
	270	Total trials

Trials were blocked by keyboard type; other factors were random within a block. The order of keyboards was balanced using a Latin square across participants to counteract systematic learning effects. The phrases to type were selected randomly so that all keyboard types saw the same phrases, although not for the same size allocations. Using standard text entry evaluation methodology (e.g., [1]), we collected completion time S (from entering the first character to the last), the input stream IS, and the transcribed text T per trial.

Procedure

Prior to starting each experimental block, the experimenter described how to use the keyboard, first using a paper model and then on the tabletop. The participants were then invited to practice using the keyboard in a series of training trials. During this time, the participant was allowed to ask questions. When participants indicated that they were ready, they were allowed to perform the actual block.

Trials were interleaved with intermission screens. Participants could only proceed to the next trial by tapping a specific target—this ensured that continuing to the next trial was a conscious decision, and also that the participant's hands were in a neutral position.

Results

The no error correction task (T1) involved typing the phrase without being able to correct mistakes. We used completion time to calculate the words per minute (WPM) values [1], and the transcribed and presented texts to calculate MSD error[3] (Figure 2 and 3).

Table 1 summarizes the significant effects on typing speed and correctness calculated using repeated-measures ANOVA. A Tukey HSD test ($p < .05$) shows that soft keyboards (SK) are significantly faster than both pinpoint keyboards (PK) and radial keyboards (RK), and that PK is faster than RK. There was no significant difference in error rate between any of the text entry methods.

		T1		**T2**
Factors	DF	WPM	Error rate	WPM
Keyboard (K)	2, 28	**209.99	0.39	**166.69
Size (A)	2, 28	**34.40	**14.91	**62.42
K * A	4, 56	**10.93	*2.92	**4.86

= $p \leq 0.05$, ** = $p \leq 0.001$.

Table 1: Effects of factors (F) on speed and errors.

In the forced task (T2), participants were required to correct their input to match the presented text. Again, Table 1 gives results from ANOVAs on the effects of our experimental conditions on typing speed (WPM). Results (Figure 2(b)) indicate that SK is again significantly faster, and that PK is faster than RK (Tukey HSD, $p < .05$).

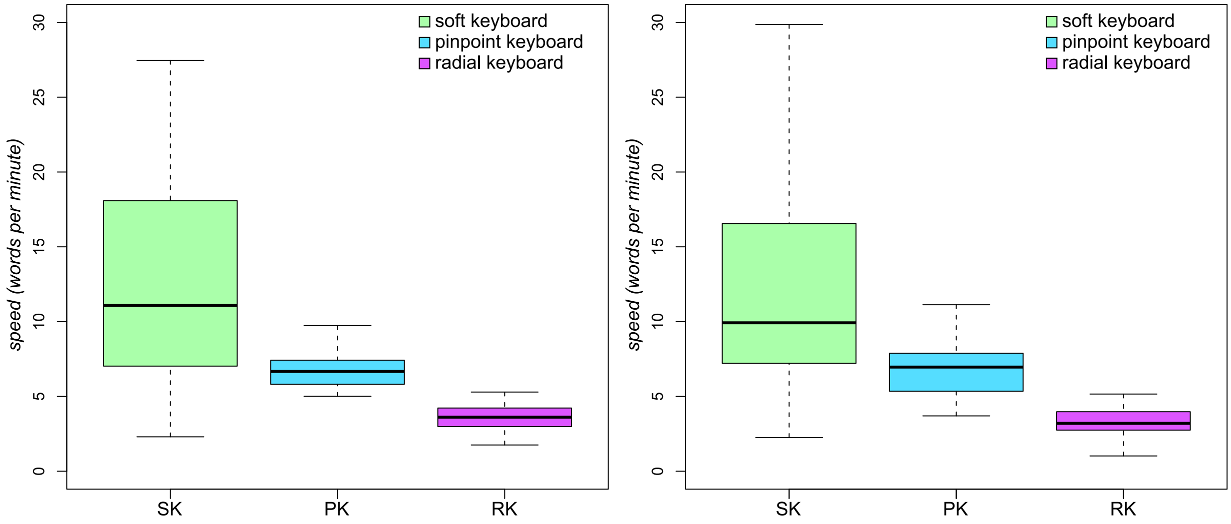
Finally, the interaction effect between keyboard and size is interesting (Figure 3). A Tukey HSD posthoc test ($p < .05$) shows that PK has significantly smaller MSD error rate than both other techniques for the small size, so is more accurate than the others (which is in line with our design goals).

Discussion

These results show that of the three text entry techniques, the standard soft keyboard is hands-down the fastest of them all. This is also a significant difference. The radial keyboard seems to translate poorly from stylus input on a mobile device to direct touch on a tabletop. There are several possible reasons for this: for example, the radial keyboard promotes continuous strokes for fast text entry [24], but perhaps the fat finger problem is exacerbated for this kind of dynamic steering task, or maybe the friction between fingertip and tabletop

---

[3] The *minimum string distance (MSD) error rate* [34] is defined as the ratio of typing errors for a given text length described in terms of the minimum string distance—the number of insertions, deletions, or substitutions separating two strings—between the presented and transcribed string.



(a) Typing speed (T1). (b) Typing speed (T2).

Figure 2: Participant typing speed (WPM) for the initial user study.

surface interferes with the interaction (the stylus has negligible friction with a display). It is also possible that the alphabetical layout had an impact—both of the other two methods used the standard QWERTY layout. In this work, we discard radial keyboards from further investigation, but this does not mean that radial keyboards are infeasible for tabletop text entry. Rather, future work is needed to study alternate radial keyboard designs that could overcome these issues.

However, the results are not as conclusive when it comes to the error rate of the other two keyboards. Furthermore, we have not yet investigated the impact of occlusion-reduction methods, such as Shift [36], that attempt to alleviate the fat finger problem. And finally, while Arif and Stuerzlinger [1] show that the error conditions used in the tasks in this experiment (none and forced, using their terminology) do not have a significant impact on text entry performance, we would still like to compare soft and pinpoint keyboards under a more natural error condition—recommended [1]—where the participant enters text normally and can correct mistakes.

To study these issues in more depth, we design a follow-up user study guided by results from the initial study.

FOLLOW-UP USER STUDY

The purpose of our follow-up study was to further investigate the performance of soft keyboards in comparison to the multitouch pinpoint keyboard in a more natural task. We also wanted to measure the impact of the Shift [36] technique, which is representative of occlusion-reduction methods commonly employed for alleviating the fat finger problem for mobile devices. Results from our initial evaluation allow us to select a single size allocation (the smallest one), as well as discard one of the mobile text entry techniques (radial keyboards) from consideration. The apparatus and procedure was the same for this experiment as before.

Participants

Fifteen paid adult volunteers (10 males, 5 females) participated in the experiment. No participant in this experiment was also a participant in the previous experiment. Ages ranged from 22 to 27 (average 24.5, median 25). All participants were experienced, although not professional, typists. Three participants were left-handed.

Task

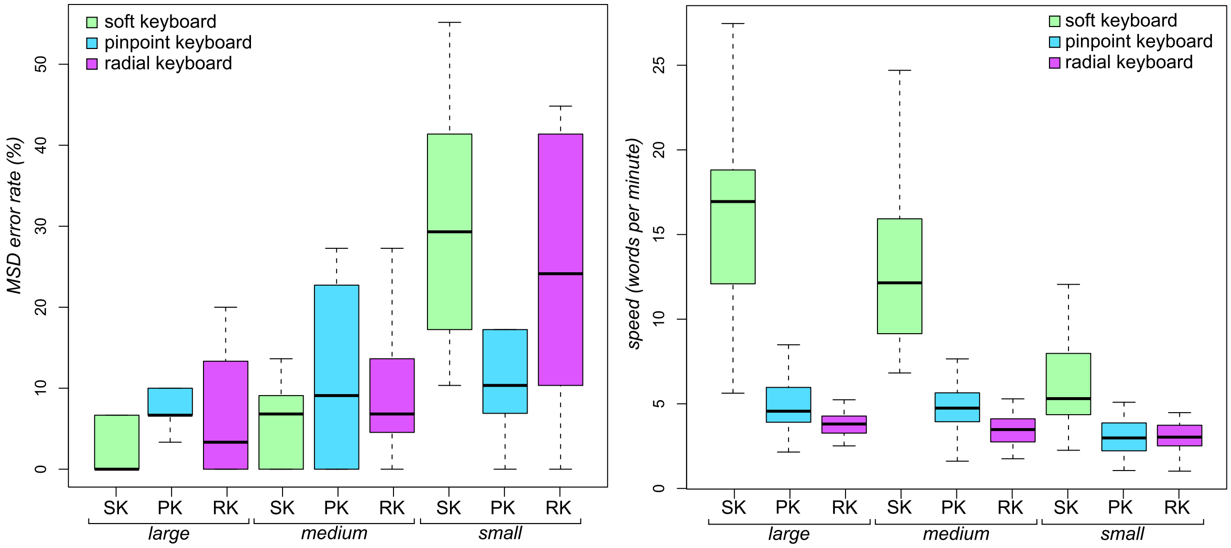
We used the recommended [1] error condition proposed by Soukoreff and MacKenzie [34], where participants can type naturally without being forced to either correct errors (T2), or not being able to correct them at all (T1). Instead, participants can make mistakes and correct them as they see fit—the error metric can cope with both fixed and unfixed errors. We used the same phrases as before.

Experimental Conditions

In this study, we included only one factor; the Keyboard Type (*K*):

- **Soft Keyboard (SK):** The standard virtual keyboard used in the previous experiment (Figure 1(a)).
- **Pinpoint Keyboard (PK):** The pinpoint keyboard used in the previous experiment (Figure 1(c)).
- **Soft Keyboard w/ Shift (SH):** A virtual keyboard with an implementation of the Shift [36] technique which creates a callout above each key to show the screen contents beneath the user's finger. The callout shows the letter on the key and is only visible when the user is touching the display. This is similar to the Apple iPhone, and gives visual feedback so that the user can move the finger before releasing the touch if the wrong key was mistakenly pressed on the initial touch. Sliding the finger on the keyboard will create callouts for any other keys that happen to become occluded during this operation.

For the size, we used the Small level (58 cm^2 or 130 × 50 pixels) from the initial study. This size was where we saw most separation in error rate between PK and SK conditions



(a) MSD error rate (T1).

(b) Typing speed (T1 + T2).

Figure 3: Impact of size allocation on error rate (MSD) and typing speed (WPM) for the initial user study.

(Figure 3). Pilot testing with even smaller size allocations led to very high error rates and highly frustrated participants, so these sizes were rejected.

Design

This experiment was also a full factorial within-participant design with the following factors:

	15	Participants
×	3	**Keyboard Types** *K* (*Standard*, *Pinpoint*, *Shift*)
×	12	Repetitions
	540	Total trials

As in the initial study, trials were blocked by keyboard type, the order was counterbalanced, phrases were randomly selected from a set, and we collected the same metrics as before.

Results

We collected results for all participants across all conditions and computed the typing speed (WPM) as well as Total Error Rate [34]—see Figure 4(a) and 4(b) for our results.

A repeated-measures analysis of variance yields a significant main effect of Keyboard type K on speed (WPM): $F(2,28) = 108.46, p < .001$. A posthoc Tukey HSD test shows that pinpoint keyboard (PK) is significantly slower than both soft keyboard (SK) and soft keyboard with Shift (SH) ($p < .001$). However, there is **no** significant difference between SK and SH ($|t| = .07, p = .9976$).

Furthermore, RM-ANOVA yields no significant main effect of Keyboard on error rate: $F(2,28) = 0.8612, p = .4303$.

IMPLICATIONS FOR DESIGN

Summarizing our results from the two user studies, we can come to the following conclusions:

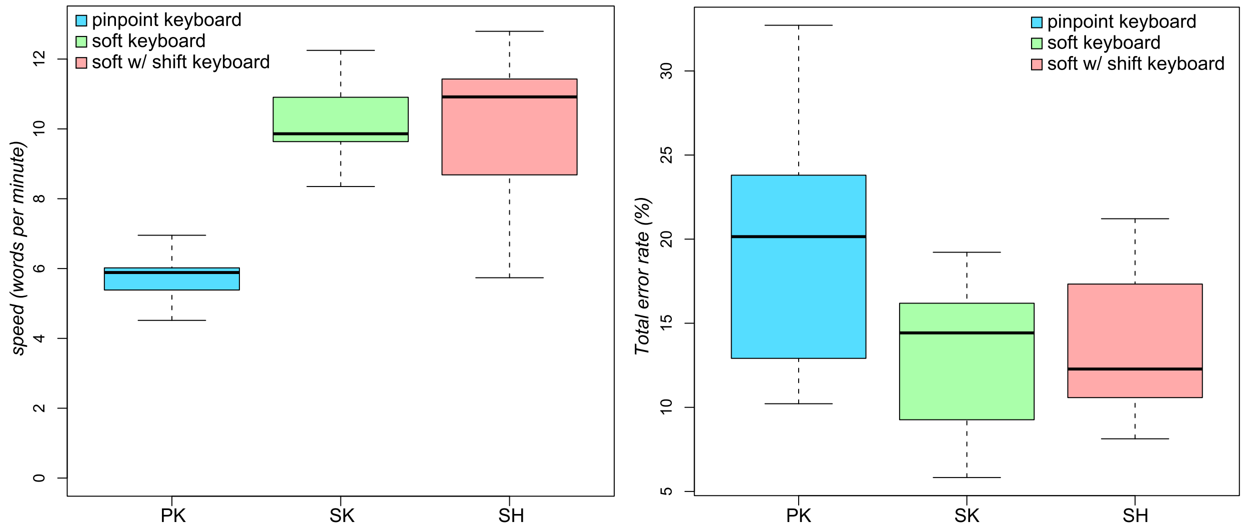
- Standard QWERTY soft keyboards allow for the fastest typing speed on tabletops in general;
- Radial keyboards optimized for mobile devices may not transfer well to tabletops without further design iterations;
- Pinpoint typing was more promising than radial keyboards but did not outperform soft keyboards; and
- The Shift [36] occlusion-reduction technique did not have a significant impact on error rate.

In this section, we will discuss the design implications for collaborative tabletop applications these results will have.

Explaining the Results

Our results raise a number of questions, the primary one naturally being whether any other technique than standard QWERTY soft keyboards is viable for text entry on collaborative tabletops. In both studies, soft keyboards exhibited both the fastest typing speed, and was not significantly less accurate than alternatives. In general, standard QWERTY soft keyboards are faster because of their familiarity, the possibility to use both hands, and the fact that mistakes are cheap. However, finger occlusion may still be a problem—particularly for small keys—and is exacerbated by the lack of tactile feedback, which also makes tapping keys more error-prone.

We adapted radial keyboards from mobile computing, but our results indicate that the method is unsuitable for tabletops. Radial keyboards were significantly slower than other techniques, yet without significantly improved accuracy. One reason may be the imprecise affordance of fingers for drawing [29], or the high occlusion [15] of using your finger instead of a thin stylus to interact with the technique. Furthermore, the friction between fingertip and tabletop surface may have made stroking motions difficult, thereby impacting the speed of text entry. In fact, the impact of friction on touch interaction is a small and hithertho underexplored aspect of tabletop surface design. We should also note that we used the default alphabetic character order, but that more optimized orders take relative letter frequency into account. Future work is necessary to determine how much key layout



(a) Typing speed (WPM).

(b) Total error rate.

Figure 4: Participant performance for the follow-up user study.

will affect this technique. While we chose to not study radial keyboards further in this work, it is entirely possible that improved radial designs may yield much better performance.

As mentioned above, our results suggest that pinpoint typing, while somewhat promising due to its low error rate in the initial user study, is still not a viable alternative to standard soft QWERTY keyboards on tabletop displays—not even on small size allocations. This seems to indicate that our initial design constraints did not have such a significant impact on tabletop text entry as we had anticipated. This is interesting in many regards—particularly in terms of the impact of fat finger occlusion on the text entry task.

More specifically, the Shift [36] technique, which can be regarded as representative of general occlusion-reduction technique because of its ubiquity in commercial products like the Apple iPhone, did **not** yield significantly lower error rate or higher speed in the follow-up user study. We believe that this phenomenon could be explained by the fact that finger occlusion clearly is governed by the physical size of keys, and not their size in screen pixels. In other words, despite our keyboards being small in screen resolution (any smaller and it would have been difficult to fit readable characters on the keys), they were still relatively large in the physical world, reducing finger occlusion to merely a minor issue.

Generalizing the Results

It can be argued that optimizing text entry methods for size is counterproductive on large displays like tabletops, and that the number of collaborators will never reach a point where this becomes important. However, screen space remains a scarce commodity even on large displays given that the view of the data should be maximized and uncluttered by interfaces [13]. Therefore, our work fills an important gap for both tabletops as well as for any large display.

In fact, one of the primary motivations for this work is the diminishing screen size available for on-screen keyboards. Our results show that this factor has a significant impact on typing performance and reaffirms the need for studying this issue in the future.

While radial and pinpoint typing did not outperform soft keyboards in our particular evaluation, this also does not mean that the tabletop text entry debate is settled. In particular, it certainly does not preclude other text entry designs from being developed and potentially yielding better performance than the current state of the art. Much work remains to be done here.

Having said that, an optimal text entry method should probably not rely on any one method in particular, but perhaps incorporate elements of several methods in combination. For example, there is nothing to prevent a QWERTY soft keyboard from having pinpoint activation areas, allowing the user to pinpoint keys instead of tapping them directly whenever the keyboard is small. Hybrid pinpointing and tapping is also possible. Furthermore, even though our results did not show a significant impact for the Shift technique, Shift only has a very small practical impact on a keyboard, and so it certainly cannot hurt to include it for tabletop text entry.

Finally, the key and somewhat surprising finding of this work has been the fact that occlusion-reduction techniques that were developed primarily for mobile device settings did not yield significantly better performance for our tabletop setting. As discussed above, this is likely due to the low ratio of screen resolution per surface units (i.e., DPI) for our tabletop display, rendering the fat finger problem a mere nuisance. Our tabletop display is not atypical: current tabletops have a long way to go until they have the same DPI as a typical mobile device. Therefore, we submit that occlusion-reduction techniques like Shift will not result in as significant improvements on tabletop displays as they have on mobile devices until tabletop DPI has reached a comparable level. This is also the most important finding in this work.

General Pinpointing

While the pinpointing technique did not turn out to be a good match for tabletop text entry, we believe that the idea of decoupling horizontal and vertical input may be interesting for other applications on multitouch displays where precision is important. We anticipate exploring pinpointing for applications like picking colors, selecting objects, or navigating the display. In particular, we think that pinpointing may be extremely powerful for collaborative tasks where occlusion is not merely an issue for the user performing the actual operation, but for other participants gathered around the table who wants to see what that person is doing on the display.

CONCLUSION AND FUTURE WORK

We have evaluated the application of popular soft keyboard-based text entry methods for mobile devices to the tabletop domain. Limited screen space allocation is an increasingly important design constraint as more participants gather around the table for collaboration. Our evaluation consists of two user studies, one where we determine suitable techniques to compare, and the other where we perform an in-depth comparison of those techniques. Our results indicate that standard QWERTY soft keyboards are significantly faster than competing techniques at the cost of no significant difference in error rate. Surprisingly, the Shift occlusion-reduction technique that is popular on mobile devices did not yield better performance. We hypothesize that this is due to the low DPI of current tabletop and large displays, which means that the targets, albeit small in screen pixels, still are large in the physical world.

Tabletop text entry is still in its infancy, and much research remains to be done. Therefore, our future work will focus on studying additional text entry methods for tabletop displays. An alternative research direction would be to study ways to perform text entry without typing altogether, such as dictionary-based methods, click-to-tag annotation, and possibly speech input for text entry. We also anticipate exploring additional applications of pinpointing.

REFERENCES

1. Ahmed Sabbir Arif and Wolfgang Stuerzlinger. Analysis of text entry performance metrics. In *Proceedings of the IEEE Symposium on Human Factors and Ergonomics*, pages 100–105, 2009.
2. Sara A. Bly. A use of drawing surfaces in different collaborative settings. In *Proceedings of the ACM Conference on Computer-Supported Cooperative Work*, pages 250–256, 1988.
3. Paul H. Dietz and Darren Leigh. DiamondTouch: A multi-user touch technology. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 219–226, 2001.
4. Collin Drury and Errol Hoffman. A model for movement time on data-entry keyboards. *Ergonomics*, 35(2):129–147, 1992.
5. August Dvorak, Nellie Merrick, William Dealey, and Gertrude Ford, editors. *Typewriting Behaviour*. American Book Company, New York, USA, 1936.
6. Leah Findlater, Jacob O. Wobbrock, and Daniel Wigdor. Typing on flat glass: examining ten-finger expert typing patterns on touch surfaces. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 2453–2462, 2011.
7. David Goldberg and Cate Richardson. Touch-typing with a stylus. In *Proceedings of the ACM INTERCHI Conference on Human Factors in Computing Systems*, pages 80–87, 1993.
8. Scott A. Golder and Bernardo A. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, April 2006.
9. Saul Greenberg and David Marwood. Real time groupware as a distributed system: Concurrency control and its effect on the interface. In *Proceedings of the ACM Conference on Computer-Supported Cooperative Work*, pages 207–217, 1994.
10. Jefferson Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 115–118, 2005.
11. Chris Harrison and Scott E. Hudson. Providing dynamically changeable physical buttons on a visual display. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, pages 299–308, 2009.
12. Björn Hartmann, Meredith Ringel Morris, Hrvoje Benko, and Andrew D. Wilson. Augmenting interactive tables with mice & keyboards. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 149–152, 2009.
13. Uta Hinrichs, Mark S. Hancock, M. Sheelagh T. Carpendale, and Christopher Collins. Examination of text-entry methods for tabletop displays. In *Proceedings of the IEEE Workshop on Tabletop Displays*, pages 105–112, 2007.
14. Uta Hinrichs, Holly Schmidt, Tobias Isenberg, Mark Hancock, and Sheelagh Carpendale. BubbleType: Enabling text entry within a walk-up tabletop installation. Technical report, University of Calgary, 2008.
15. Johannes Hirche, Peter Bomark, Mikael Bauer, and Pawel Solyga. Adaptive interface for text input on large-scale interactive surfaces. In *Proceedings of the IEEE Workshop on Tabletop Displays*, pages 153–156, 2008.
16. Petra Isenberg and M. Sheelagh T. Carpendale. Interactive tree comparison for co-located collaborative information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1232–1239, 2007.
17. Per-Ola Kristensson and Shumin Zhai. SHARK^2: a large vocabulary shorthand writing system for pen-based computers. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 43–52, 2004.

18. I. Scott MacKenzie and R. William Soukoreff. Text entry for mobile computing: Models and methods, theory and practice. *Human-Computer Interaction*, 17(2/3):147–198, 2002.

19. I. Scott MacKenzie and R. William Soukoreff. Phrase sets for evaluating text entry techniques. In *Extended Abstracts of the ACM CHI Conference on Human Factors in Computing Systems*, pages 754–755, 2003.

20. I. Scott MacKenzie and Shawn X. Zhang. The immediate usability of Graffiti. In *Proceedings of Graphics Interface*, pages 129–137, 1997.

21. I. Scott MacKenzie and Shawn X. Zhang. The design and evaluation of a high-performance soft keyboard. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, pages 25–31, 1999.

22. I. Scott MacKenzie and Shawn X. Zhang. An empirical investigation on the novice experience with soft keyboards. *Behaviour and Information Technology*, 20(6):411–418, 2001.

23. I. Scott MacKenzie, Shawn X. Zhang, and R. William Soukoreff. Text entry using soft keyboards. *Behaviour and Information Technology*, 18(4):235–244, 1999.

24. Jennifer Mankoff and Gregory D. Abowd. Cirrin: A word-level unistroke keyboard for pen input. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 213–214, 1998.

25. Donald A. Norman and Diane Fisher. Why alphabetic keyboards are not easy to use: Keyboard layout doesn't much matter. *Human Factors*, 24(5):509–519, 1928.

26. Ken Perlin. Quikwriting: Continuous stylus-based text entry. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 215–216, 1998.

27. Nathaniel Rochester, Frank C. Bequaert, and Elmer M. Sharp. The chord keyboard. *Computer*, 11(12):57–63, December 1978.

28. Yvonne Rogers and Sian E. Lindley. Collaborating around vertical and horizontal large interactive displays: Which way is best? *Interacting with Computers*, 16(6):1133–1152, 2004.

29. Kathy Ryall, Clifton Forlines, Chia Shen, Meredith Ringel Morris, and Katherine Everitt. Experiences with and observations of direct-touch tabletops. In *Proceedings of the IEEE Workshop on Tabletop Displays*, pages 89–96, 2006.

30. Toni Schmidt, Werner A. König, and Harald Reiterer. Text input on multitouch tabletop displays. In *Poster Proceedings of the IEEE VGTC/Eurographics Symposium on Visualization*, 2009.

31. Andrew Sears, Julie A. Jacko, Josey Chu, and Francisco Moro. The role of visual search in the design of effective soft keyboards. *Behaviour and Information Technology*, 20(3):159–166, 2001.

32. Andrew Sears, Doreen Revis, Janet Swatski, Rob Crittenden, and Ben Shneiderman. Investigating touchscreen typing: The effect of keyboard size on typing speed. *Behaviour and Information Technology*, 12(1):17–22, 1993.

33. R. William Soukoreff and I. Scott MacKenzie. Theoretical upper and lower bounds on typing speed using a stylus and a soft keyboard. *Behaviour and Information Technology*, 14(6):370–379, 1995.

34. R. William Soukoreff and I. Scott MacKenzie. Metrics for text entry research: an evaluation of MSD and KSPC, and a new unified error metric. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, pages 113–120, 2003.

35. Dan Venolia and Forrest Neiberg. T-cube: A fast, self-disclosing pen-based alphabet. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, pages 265–270, 1994.

36. Daniel Vogel and Patrick Baudisch. Shift: a technique for operating pen-based interfaces using touch. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, pages 657–666, 2007.

37. Malte Weiss, Julie Wagner, Yvonne Jansen, Roger Jennings, Ramsin Khoshabeh, James D. Hollan, and Jan O. Borchers. SLAP widgets: bridging the gap between virtual and physical controls on tabletops. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, pages 481–490, 2009.

38. Daniel Wigdor, Gerald Penn, Kathy Ryall, Alan Esenther, and Chia Shen. Living with a tabletop: Analysis and observations of long term office use of a multi-touch table. In *Proceedings of the IEEE Workshop on Tabletop Displays*, pages 60–67, 2007.

39. Shumin Zhai, Michael A. Hunter, and Barton A. Smith. The Metropolis keyboard - an exploration of quantitative techniques for virtual keyboard design. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 119–128, 2000.

40. Shumin Zhai and Per-Ola Kristensson. Shorthand writing on stylus keyboard. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, pages 97–104, 2003.

41. Shumin Zhai, Per Ola Kristensson, and Barton A. Smith. In search of effective text input interfaces for off the desktop computing. *Interacting with Computers*, 17(3):229–250, 2005.

42. Shumin Zhai, Alison E. Sue, and Johnny Accot. Movement model, hits distribution and learning in virtual keyboarding. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, pages 17–24, 2002.