RearType: Text Entry Using Keys on the Back of a Device

James Scott, Shahram Izadi
Microsoft Research
Cambridge, UK
{jws,shahrami}@microsoft.com

Leila Sadat Rezai
RWTH Aachen
Germany
Leila.Rezai@rwth-aachen.de

Dominika Ruszkowski, Xiaojun Bi, Ravin Balakrishnan
Dept. of Computer Science,
University of Toronto, Canada
{druszkow, xiaojun, ravin}@dgp.toronto.edu

ABSTRACT

RearType is a text input system for mobile devices such as Tablet PCs, using normal keyboard keys but on the reverse side of the device. The standard QWERTY layout is split and rotated so that hands gripping the device from either side have the usual keys under the fingers. This frees up the front of the device, maximizing the use of the display for visual output, eliminating the need for an onscreen keyboard and the resulting hand occlusion, and providing tactile and multi-finger text entry – with potential for knowledge transfer from QWERTY. Using a prototype implementation which includes software visualization of the keys to assist with learning, we conducted a study to explore the initial learning curve for RearType. With one hour's training, RearType typing speed was an average 15 WPM, and was not statistically different to a touchscreen keyboard.

Categories and Subject Descriptors

H.5.2 **[User Interfaces]:** Input devices and strategies (e.g., mouse, touchscreen); Prototyping

General Terms

Human Factors

Keywords

Text entry, keyboard, mobile devices, tablet PC.

1. INTRODUCTION

Despite the increasing popularity of larger form factor mobile devices such as Tablet PCs, Ultra-Mobile PCs (UMPCs), e-ink based devices (e.g., Amazon's Kindle), or the Apple iPad, enabling effective methods for text entry on these devices remains a difficult problem. Techniques that utilize direct on-screen input with a pen or touch rarely approach the text entry speeds of a regular keyboard, and perhaps more importantly suffer from the problem where the user's hands significantly occlude screen content, often requiring contortions of the hand and fingers to achieve a workable tradeoff between reading the screen and entering text [20]. Regular physical keyboards obviously enable much faster entry speeds, but fitting them onto these devices while maintaining usability in mobile scenarios remains an industrial design challenge.

Drawing inspiration from recent work on using the back of a device [1, 21] as a continuous interaction space, we developed RearType (Figure 1) which is a text entry approach that works by placing discrete physical keys on the back of devices where fingers gripping the device on either side can reach them. Our goal is a system that provides the tactile feedback and familiarity of a regular keyboard without cluttering the front of the display, ameliorates the occlusion problem inherent in direct on-screen touch and pen input, does not use the valuable screen real-estate taken up by an on-screen keyboard, leverages users existing skills in touch-typing on a regular physical QWERTY keyboard, and allows for text entry in highly mobile usage scenarios.

The RearType key layout takes the two halves of the QWERTY alphabet keys and rotates them so that the keys remain in the same relative positions with respect to the fingers despite them being on the rear surface of a device (Figure 1). This is equivalent to 90-degree rotations in two axes. The thumbs remain on the front surface for grip, and other keys are placed within reach of the thumbs.

RearType avoids the need for on-screen touch-based keyboards and their inherent space usage and occlusion problems. It also supports 10-finger touch-typing input versus the single-finger "hunt and peck" of touchscreen or thumb-based small-key keyboards (e.g. on Blackberrys). RearType is also well-suited for scenarios where there is no natural surface on which to rest a device, such as when standing or reading on a sofa, as the hands grip the device as well as typing, unlike other 10-finger typing keyboards on mobile devices.

With a new technique like RearType that varies from the status quo in several important design dimensions, the challenge in evaluation is in determining which dimensions to consider first. After extensive discussion and pilot explorations, we felt that the crucial question to ask at this stage is whether the basic idea of putting keys on the back of the device is even reasonable. After all, at first glance, this might well appear to be a crazy idea with little practical appeal. We therefore conducted a user study to determine if, upon initial exposure to the device, users find such a novel keyboard at all usable or simply frustrating. To obtain some comparative empirical data, we also contrasted the use of RearType with a touchscreen soft keyboard and a regular physical keyboard. Initial results are very promising; we show that after only an hour with the device, participants could type 15 WPM using RearType and that their performance was not statistically different to text entry using the touchscreen keyboard.

In addition to the introduction of the RearType concept and initial evaluation study, this paper also contributes in the design and implementation of prototype RearType hardware and visualization

MobileHCI 2010 September 7 - 10, 2010, Lisbon, Portugal.
ACM 978-1-60558-835-3/10/09.

(a) Front of RearType prototype

(b) Back of RearType prototype

Figure 1. RearType prototype (a) front, with thumb buttons, showing study software including semi-transparent visualization (the 'h' key has just been pressed), and (b) back, with keys in rotated QWERTY layout (fingers lifted from home keys for better view).

software to enable novice users to learn the RearType layout. Drawing from our experience in prototyping RearType and our user study, we also make recommendations for future RearType devices.

2. RELATED WORK

Mobile device text entry has been extensively studied [18]. We summarize the work most relevant to RearType here.

Many "ergonomic" keyboards keep the QWERTY layout but split and angle the keys. For example, SafeType [15] rotates the keys so they are vertical – this is one of the two rotations applied to get the RearType layout.

The Half-QWERTY keyboard [14] enables one-handed typing using only half the normal QWERTY keys and the spacebar to switch halves, and the use of QWERTY facilitates a quick transfer of skill, with subjects reaching 50% of their full QWERTY speed in around 8 hours. Green et al. [6] have prototyped a reduced QWERTY keyboard where only a single row is present, with significant typing speeds achieved after only a few minutes training: 10 WPM for multitap-based disambiguation.

The Grippity see-through keyboard enables "BackTyping" [7], i.e. typing with the fingers on the rear using a standard QWERTY layout, and the see-through nature means it avoids the need for a software visualization to assist users finding keys. However, unlike RearType, it does not use physical tactile keys, and the see-through keyboard requires space on the front of the device.

LucidTouch [21] allows touch input on the back of small devices, using the illusion of semitransparency to enable accurate targeting. However, for text entry the screen would still have to have an on-screen keyboard. BehindTouch [9] is similar to the LucidTouch system but with touch sensing rather than touch+hover sensing. NanoTouch [1] is a significantly smaller-scale version of LucidTouch and introduces some nice techniques that actually turns the "fat finger" problem into an advantage, but does not address text input per se. HybridTouch [19] uses both a pen on the front and touch on the rear of a device.

RearType and chording keyboards such as the Twiddler [12] are similar in that they both use front and rear keys, and that the keying hand(s) also grip the device. However, RearType uses only a single keypress per lowercase character, and the use of the standard QWERTY layout makes its operation more obvious "at a glance" (though not necessarily quicker to learn). On the other hand, chording keyboards are easier to integrate in smaller form factor devices such as mobile phones, where there may not be enough room for RearType. Chording keyboards can be faster than traditional keyboards [3], and other work has addressed the optimal mapping of chordings to characters [5]. Future designs of RearType could leverage some of these findings.

Wobbrock et al [23] compared thumb and index finger touch input on the rear and front of devices, and found the index finger to work as well on the rear as the front, but the thumb to be less dexterous on the front than expected. While at a high level the former result is encouraging and the latter is a concern for RearType, those results may not actually transfer well to our situation where physical keys are used. Wobbrock and others also compared front and rear gestural text input using an isometric joystick [22], finding that text input was performed using a rear joystick at 70% of the speed of the front joystick.

Other studies have looked at "blind" use of mobile keypads [17], finding that tactile feedback is crucial to performance (providing justification to RearType's use of physical keys), and blind use of Mini-QWERTY/chording keyboards [2] showing that accuracy was worse on Mini-QWERTY than chording keyboards. While the relatively poor performance of Mini-QWERTY might give us pause, such keyboards are operated with just two thumbs while RearType uses all ten fingers.

Another method of mobile text input is stylus input and handwriting recognition, with speeds of 21-25 WPM with modest practice [10]. However, handwriting input also causes the user to occlude the screen while input on the rear leaves the full screen viewable.

3. PROTOTYPE

In order to test the feasibility for this novel mode of mobile keyboard entry, we built a prototype device using the following components:

1) A Multi Function Panel from CH Products (www.chproducts.com). This provides a flat 28cm by 24cm base, onto which individual keyboard keys can be arbitrarily attached. Key press events are transmitted to a connected PC

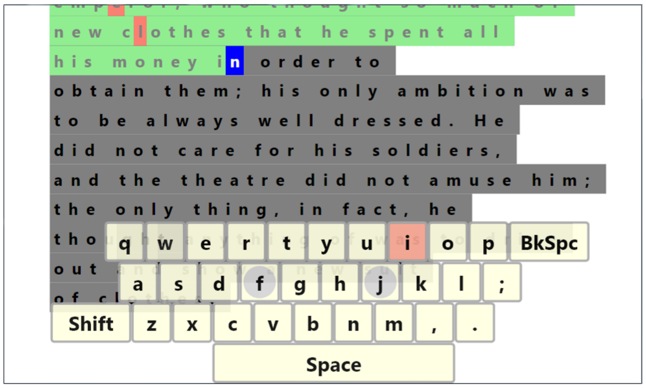


Figure 2. Study software showing typed text (green with red errors), untyped text (grey) and the current character (blue). The lines scroll upwards so the current line remains near the top of the screen. This is the "touchscreen" condition, so the visualization is in QWERTY layout and it accepts touch input. A recently touched key glows red.

via USB. The product is used for configurable keyboards, particularly for gaming, but provides a useful platform to experiment with different key arrangements.

2) A Mimo 740 7-inch 800x480 peripheral screen with a resistive touchscreen, also connected via USB.

3) A Windows Vista PC running custom software for visualization and our study software (written in C# for Microsoft .NET Framework 3.5).

The Multi Function Panel was modified by removing some of the foil shielding from its back (which we use as the front) – see Figure 1(a). This allowed us to place some keys on the front of the device near the edges. The Mimo screen was modified by laser-cutting the back panel to remove the mounting protrusion, allowing it to be velcroed to the Multi Function Panel more closely. The manufacturer-provided software drivers were used for both devices. We applied some sticky-back neoprene to the sides of the device to improve the users' grip.

3.1 Key Layout

In initial explorations with the device we empirically confirmed two intuitive facts:

1) The need to grip the device and press keys restricts movement of the fingers in comparison to regular physical keyboards. With keyboards, the wrist commonly acts as a pivot point, and to hit some keys it may be lifted, (e.g. Escape). With RearType, the palm of the hand is the pivot, and it is hard to "lift" this since this would result in the opposite hand supporting the whole device.

2) The thumbs are naturally placed on the front of the device, and can be used to press a small set of keys mounted on the front (See Figure 1(a)).

We then experimented with various key layouts and eventually settled on the following layout, which is depicted in Figure 1(b).

The rear has 3 lines of 5 keys per hand (e.g. "QWERTY"), so there are 30 keys on the back in total. A fourth line (e.g. for numbers) proved too hard to reach and was not included. In our prototype the B key is duplicated because informal prior experience with split keyboards showed that B was hit with either hand (it is the only alphabet key equidistant from both home keys).

On the front, we placed other commonly used keys: Space, Backspace, Enter and Tab in a column under the right thumb's natural position when gripping the device, and the modifiers Shift, Control and Alt under the left thumb similarly. However there is not enough room to place all the remaining keys individually on the front, both because of reachability by the thumbs, and because placing too many keys on the front defeats the purpose of maximizing space for the screen and the very rationale for RearType.

We therefore added a "Mode" key to provide further functionality to the other keys. For example, the home row of rear keys (ASDFGHJKL;) is used for the numbers (1234567890) when Mode is held down. Furthermore, Shift and Mode work orthogonally, with Shift retaining the functionality from the normal keyboard, so that "Shift-Mode-A" becomes "Shift-1" which is "!". This design allows users to draw on their existing knowledge of Shift, only having to learn Mode. While Shift and Mode could be placed under opposite thumbs, we also found that a single thumb can press both keys simultaneously if they are placed next to one another, allowing the other thumb to be used for a third key if necessary.

Our use of a Mode function is similar to the way that existing touchscreen keyboards work, e.g., the iPhone keyboard. While the "moded" keys may be harder to learn since there is less transfer of skill/knowledge from standard keyboards than for alphabetic keys, and we are well aware of the problems generally associated with modes [16], we can use visualizations of the keys on the screen to help reduce this problem for users.

3.2 Visualization

To assist users in learning and using RearType, we implemented a visualization of the keys highlighting any key(s) being pressed. As with existing on-screen keyboard applications, the visualization changes automatically when Shift or Mode is pressed to show the character that would be generated by each key in that situation. In explorative studies before our main study, we trialed a number of different orientations of the visualization: *"QWERTY"* – visualizing the keyboard as if it were a normal keyboard layout rather than rotated, "*see through*" – visualizing the keys as if you could see through the device, and a "*diagonal representation"* – which was halfway between the other two. The "see through" visualization (shown in Figure 1(a)) was clearly preferred in our trial studies, so we used this in our main study.

The visualization also provides us with touchscreen keyboard functionality (when viewed in "QWERTY" mode as shown in Figure 2) which we used in the study.

3.3 Avoiding Unintentional Presses

One of the problems with putting keys on the back is that when the device is picked up or rested down, these keys may be hit or held down, resulting in unwanted keypresses. In our prototype we did not address this issue directly. However, this problem can be avoided either by adding an explicit control to enable the keys (e.g., flicking an enable/disable switch, perhaps under a thumb), or by automatic sensing of the hands or fingers being in the correct state (e.g., through capacitive or pressure based sensing underneath the grips for both hands or on the individual keys themselves [8]).

4. STUDY

As with many new input technologies, there are a large number of parameters and conditions that could be studied. For RearType this includes the ideal placement of the keys for speed; accuracy and comfort, novice user experience/performance versus expert user experience/performance; various usage settings including standing, reclining, or mobile; the need for visualization and the best visualization to use; whether there was knowledge transfer from QWERTY layouts or not; alternatives to the QWERTY layout; the extra training and speed/accuracy reduction for moded keys; and so on. However, as alluded to in the introduction, and to limit the scope of this first study of RearType, we chose to focus on two core initial issues (with others being left for future work informed by this initial work):

1) Can users learn to type English prose on the RearType keyboard after a modest amount of first exposure to the device (~1 hour)

2) After this modest amount of training, how does RearType input speed compare against a touchscreen soft keyboard (a realistic and currently-used alternative for the tablet/UMPC form factor that we designed RearType for) and how do both compare against the baseline of a normal keyboard?

4.1 Participant Selection

We found in pilot studies that a very relevant factor for how quickly RearType was learned was whether users were "10 finger" typists. Those using only a few fingers on each hand tend to move their hands over the keyboard more, and this is not easy using RearType since the palms must grip the device. We therefore tried to select for "10 finger" typists in our study, and while we acknowledge the effect of this choice on the generalization of the results, we felt it was more important at this stage to see if at least some groups of users could find RearType useful.

We recruited participants by posting on <anonymized website> offering $50 to be involved in a two-hour typing study based in <anonymized city>. We asked interested people to submit a screenshot showing a words-per-minute score from a specific prose task on an online typing tutor website (http://www.typingweb.com/), and furthermore asked them whether they (a) did not look at the keyboard and (b) typed with 10 fingers. From the 70 respondents, we rejected those who demonstrated less than 50 words per minute typing speed on the online tutor, and those who self-reported that they looked at the keyboard or were not "10 finger" typists, resulting in 45 candidates for the study. We then chose 12 participants uniformly distributed in the range of remaining speeds, and invited them for the study. We ran a single pilot (results not reported due to space limitations) and in response slightly reduced the length of the typing tests and training to target a study duration of two hours.

Of the twelve participants, eight were male and four were female (P3, P7, P9, P12); nine were aged 18-24, two were 25-34 (P2, P6) and one was 35-44 (P3). There were eight students ranging from undergrads to postgrads, an IT sysadmin, a teacher, a receptionist and a web programmer. We did not preselect on the basis of these demographics, only on typing speeds.

4.2 Study Design

In order to answer the research questions above, we ran a study in four phases.

First was a software familiarization phase, which consisted of typing one prose task using a normal keyboard. This was to familiarize the user with the study software; we assumed that no familiarization with a standard keyboard was needed.

Second was a RearType training phase. During this phase, the user was run through a typing tutor style series of lessons for RearType. This started with 6 tasks with increasing difficulty on the "home row" keys, for example the first line of text in the first task was:

```
fff jjj fff jjj ff jj ff jj f j f j
```

This trained the participant in using the Space key (which is on the front of the device) alternating with rear keys, and in hitting the home row keys. Similar sets of tasks were then provided for the top row and bottom row keys. A few more tasks introduced the use of capital letters (with the shift key on the front) and full sentences. The final training phase task was to type in a short story totaling 1300 characters, split into four tasks for manageability. In total the training phase consisted of 25 tasks and 2700 characters and was designed to last around 50 minutes.

Third was a two-task touchscreen soft keyboard familiarization phase, where the user typed a total of 657 characters of prose with the touchscreen, to familiarize them with that condition. We assumed that the user was comfortable with the general concept of a touchscreen keyboard; hence this short familiarization phase was simply to get them used to the particular touchscreen employed in the prototype.

Fourth was an experimental phase consisting of five texts drawn from the stories of Sherlock Holmes, with 343-350 characters in each text. All five texts were entered, then these five tasks were repeated for a second and third time. However, each time a single text was entered the input method cycled around between our three conditions: keyboard, touchscreen and RearType. This cycle ensured that each text is entered using each input method. We randomized the order of the input methods between the participants – with six possible orderings and twelve participants, each ordering was used twice. This method of counterbalancing was used for two reasons. First, in case the study proved too long for some participants and they did not complete it, a partial log would have contained typing tasks with all three conditions (however, this did not occur in the study – all tasks were completed). Second, to minimize fatigue by forcing users to change their input method frequently, as our study involved a long series of input tasks with a new text entry technique.

Since our software acts as a typing tutor we wished to enforce that participants typed the text correctly to gain the experience that the series of lessons was designed to give. We therefore used two measures: (a) during the training phases, if the final text (after any corrections) was under 90% accurate, the whole task had to be repeated, and (b) the user was not permitted to type two incorrect characters in a row (the second and subsequent incorrect character was ignored); thus, to make progress, the user had to pay attention to the text. After an error, the user could therefore only type either backspace or the next correct character. We kept the latter restriction into the experimental phase, both for consistency, and because it allowed us to perform key-by-key analysis more easily since long strings of errors, e.g. due to insertion of a duplicate character, were not permitted.

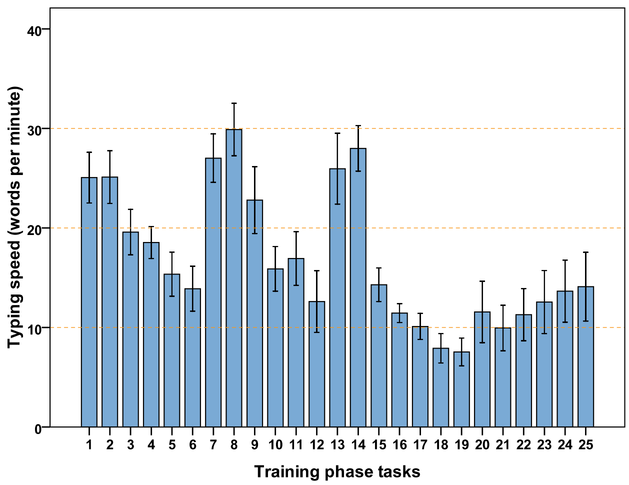


Figure 3. Mean typing speed for participants in RearType training tasks (+/- SEM). Trial index 1-6 are "home row", 7-12 are "top row", 13-18 are "bottom row", 19-21 are "caps and sentences", 22-25 are the final prose exercise.

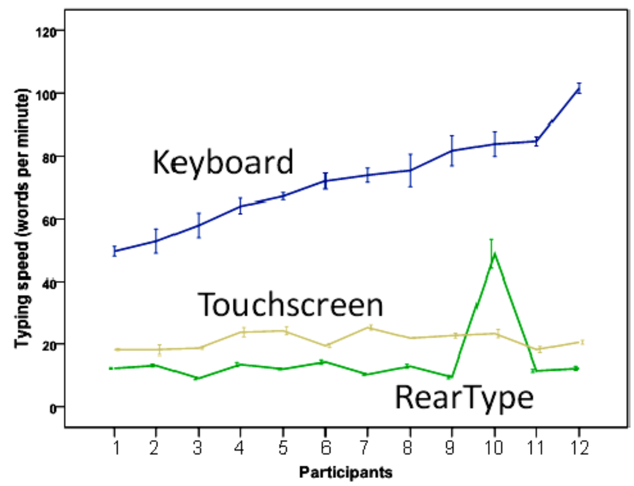


Figure 4. Mean typing speeds across all experimental phase texts, for each participant (+/- SEM). Participants are ordered by increasing keyboard input speed (this ordering is used throughout the paper).

We aimed for the experimental phase to last around 50 minutes, so the whole study totaled around 2 hours (with setup time, familiarization phases, etc). In between each task, the participant was shown their accuracy and speed for the previous task, instructions for the next task (including which type of input it required) and given an opportunity to rest before pressing a touchscreen button to continue.

To simplify this preliminary study, we restricted the study texts to only include "non moded" keys (i.e. the uppercase and lowercase alphabet, semicolon, comma and period).While this obviously does not cover the whole range of characters in many typing scenarios, we decided that (a) it would not be possible to train and test the use of further characters in just 2 hours per participant, and (b) the data we would get from this study would provide useful initial results as to the learning curve, the comparison with a touchscreen, and the design of RearType. Furthermore, in constructing the tasks, we only had to remove a few hyphens and question marks. Thus, for the texts we used, only a small amount of time would have been spent typing these characters compared to the large number of characters which were already present.

4.3 Experimental Setup

We provided users with the RearType prototype described running our study software, with on-screen instructions as well as scripted instructions from the experimenter. We also provided a stand so that the device could be placed at a comfortable reading angle in between studies. The stand was also used for the keyboard entry condition; a normal keyboard was placed in front of the stand for use in that condition. Participants sat on a chair with arms and we allowed users to support the device however they wished (for example resting it on their knees or lap) during touchscreen or RearType use.

We disabled physical keys when using the touchscreen and vice versa, to avoid false presses on the RearType and touchscreen conditions (e.g., triggering RearType keys when resting the device on lap in the touchscreen condition). We removed all the front keys (e.g. the Mode key) apart from the three required in the study (shift, space and backspace) to avoid confusion.

We also administered a pre-study questionnaire on demographics and a post-study questionnaire on participants' preferences, comfort, and their expectations of RearType if they were to use it in the longer term.

5. RESULTS

The actual study time varied greatly, from a very fast 66 minutes to 162 minutes (mean 129), due to the speed of entry using the various input methods varying (as discussed below). In total, 549 tasks were completed, of which 33 were repetitions due to the achieved accuracy being below 90%. However, none of these repetitions were in the experimental phase; 29 were RearType training tasks and 4 were touchscreen familiarization tasks.

5.1 Learning Curve

Figure 3 shows the average typing speed during the RearType training phase. Some trials (particularly the first few on each row) are easier since they contain repetitive presses with little finger movement. While speed in the final prose entry tasks is slower than the easy tasks, one hour of training was enough to enable participants to type at an average 15 words per minute (WPM)[1].

RearType speed during the experimental phase varied greatly, from 9 (P3) to 47 (P10) WPM, showing that, as expected, some people learn to use RearType more quickly than others. A real RearType typing tutor should therefore be adaptive to users advancing more quickly or slowly. In our initial software, we had incorporated a minimum WPM limit before advancing to the next task (i.e., making it adaptive), but we removed this after the pilot since it seemed this might increase the duration of the study to an unacceptably high level. In a real tutorial scenario, however, where the goal is to achieve proficiency rather than complete training as fast as possible, such adaptation should remain.

[1] We use WPM rather than Characters Per Second (CPS) in our results for comparability with other text input literature [18], where WPM is defined as CPS * 5 / 60 rather than reflecting the actual number of characters per word in the text (5.6 for our experimental texts, inclusive of spaces/punctuation).

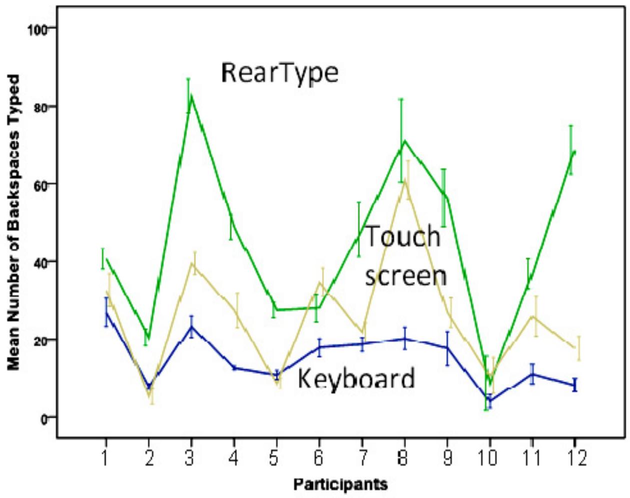


Figure 5. Mean number of uses of backspace key in experimental trials (+/- SEM)

5.2 Input Method Speed Comparison

Figure 4 compares the typing speeds of participants during the experimental phase. We order the participant numbering by their experimental performance in typing speed using a standard keyboard. As intended by our participant selection, we had a range of typing speeds from 50 to 100WPM, though this turned out to have no correlation to their speeds using the touchscreen or RearType input mechanisms.

Overall mean average across all conditions was 46.8 WPM. A repeated-measures ANOVA found a significant main effect for input technique on typing speed ($F_{(2, 22)}=121.6$ $p<0.001$). Overall, the mean speed for RearType was 15.1 WPM, touchscreen 21.2 WPM and keyboard 72.1 WPM. As expected, pair-wise comparisons of keyboard with RearType and with touchscreen shows keyboard to be significantly faster ($p<0.001$ for both). Most interesting, however, is that a pair-wise comparison of RearType and touchscreen shows no significant difference ($p = 0.069$).

Although RearType was slower than the regular keyboard (which is to be expected given all our participants' significant familiarity with the keyboard), the fact that there was no significant difference in typing speed between RearType and the touchscreen after just one hour's training is extremely encouraging. Also, note from Figure 4 that one participant (P10) showed faster RearType speed than touchscreen, with 47 WPM compared to 23 WPM for touchscreen and 83 WPM for keyboard.

5.3 User Preferences and Comfort

Immediately after the two-hour study, we asked participants to reply to a post-study questionnaire to gauge their experiences with and thoughts about the device.

Participants were asked to rank the input methods according to their perceived speed (ties permitted). All participants said that the keyboard was quickest. Surprisingly, four said that RearType was quicker than the touchscreen (including P10 who was actually quicker on RearType, as well as P5, P7 and P12 who were not), and two more said it was equally fast (P2, P4).

When asked to rank the input methods according to ease of use (no ties permitted), again all participants placed the keyboard first. Encouragingly, two-thirds (8 out of 12; P2, P5-7, P9-12) said that RearType was easier to use than a touchscreen. When directly asked to choose which they would use for text input between RearType and the touchscreen, if they owned a device which had both input mechanisms available, half the participants (P2, P4-6, P10, P12) chose RearType. While some of this might well be due to the Hawthorne effect [11], it is nonetheless encouraging.

However, in other freeform text questions, participants were less enthusiastic. While two participants (P2, P4) commented favorably on the comfort of gripping the device, nine participants (P3, P5-12) commented that the device was either uncomfortable, heavy, or hard to grip. These issues are somewhat unsurprising as this was a research prototype that had not undergone any real industrial design, and can certainly be addressed with ergonomic improvements – e.g. the sides could be molded to an easy-to-grip shape. Two participants (P1, P10) suggested that the device somehow be supported with the wrists, which would take the pressure off the palms. Similarly, the top edge of the hand could help support the weight of the device, if some protrusions from the device allowed the weight to rest there. Another option would be to move the grip location and keys to two bottom corners of the device (squaring those corners off), so that gravity rather than grip would be responsible for supporting the device to some extent.

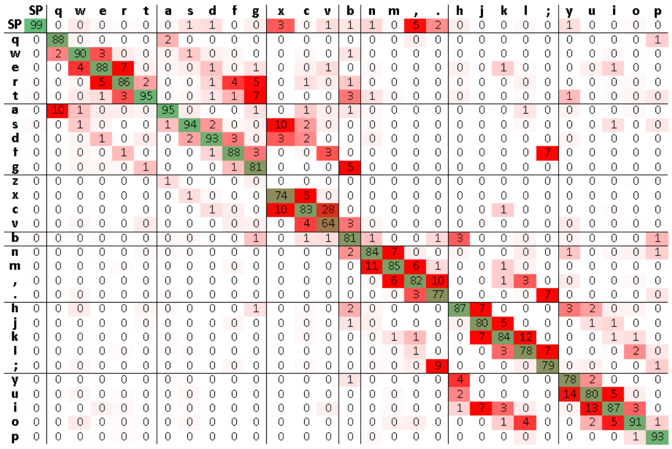
5.4 Requirement for Visualization

Although we did not turn off the visualization at any time during the trial, one of the potential advantages of RearType is the ability to use nearly the entire front surface of the device for display output, and to do this the visualization would have to be disabled once the user was a confident touch typist with RearType. It is worth noting, however, that even with visualization on, RearType improves upon a touchscreen, where the fingers occlude the display whereas RearType results in no hand occlusion and the visualization can be semi-transparent.

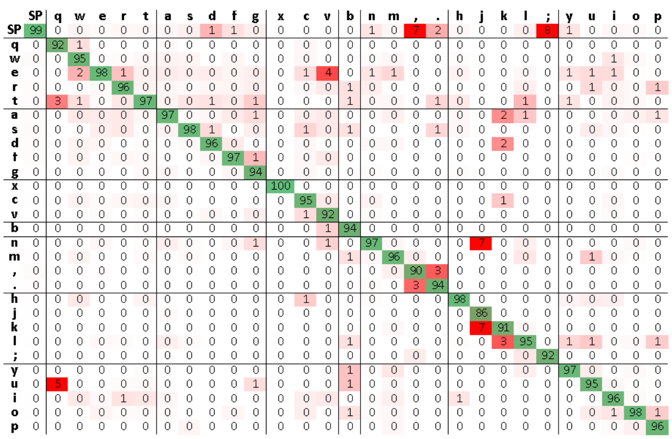
To explore the visualization issue further, we asked participants in the post-study questionnaire how often they used the visualization during the study. Seven replied "Often", three (P4, P7, P12) "Sometimes" and two (P2, P10) "Rarely/Never". We also asked users whether they agreed with the statement *"If I used it frequently, I am confident that I could use RearType for text input with the visualization turned off"* on a five-point Likert scale. One replied "Strongly Disagree" (P9), two replied "Disagree", (P4, P6), two were "Neutral" (P3, P11), six (the median) replied "Agree", and one (P10) replied "Strongly Agree". While this is forward-looking rather than based on experience, it provides some support for the idea that the screen can be freed up for output only, or the visualization can be adaptively used depending on circumstance. This is also partially supported by responses to a free-text question on what strategies people used to avoid getting stuck – seven participants (P2-3, P5-6, P8, P10, P12) reported using the home keys as anchors, while five (P2, P4, P6, P8-9) reported using the visualization (some did not specify).

There are other techniques that may also support an adaptive visualization without being too distracting to the user whilst they use regular applications, particularly given the visualization is used for output only. For example, the visualization might "fade in" under some circumstances, for example when the Mode key is pressed, (as users may take longer to learn the Mode-based characters), or when a user deletes a character (as an indication that they are hunting for a key).

We could also improve the visualization – both for training and use – by incorporating a representation of the hand positions in the



(a) RearType



(b) Keyboard

Figure 6. Key confusion matrix, showing percentages of times that the expected lowercase character (column) resulted in each actual character (row), for "advancing" keypresses (i.e. at most one error in a row) in the experimental phase. "SP" = Space.

style of LucidTouch [21], allowing users to see what keys they were about to press. This could be based on proximity sensing, allowing users to see their fingers hovering under the device, on touch sensing, allowing users to see which keys they were lightly touching before depressing them [8], or on pressure sensing, allowing users to press a key lightly to check that it is correct before pressing more firmly.

Finally, in our early testing of RearType, we found that using the visualization sometimes hindered the transfer of QWERTY skills to RearType, and this was backed up by a few of our participants' comments:

P12: "*visualization confused me. It was quite helpful to imagine myself typing on a physical keyboard without thinking too much.*"

P7: "*Stuck at first when typing letter to letter, improved when I looked at word as a whole then went faster.*"

While we have not quantified these effects in this study, it may be that the visualization should not be always present even in the early stages of learning, to encourage QWERTY skill transfer. For example, after each lesson a typing tutor could repeat the lesson without visualization.

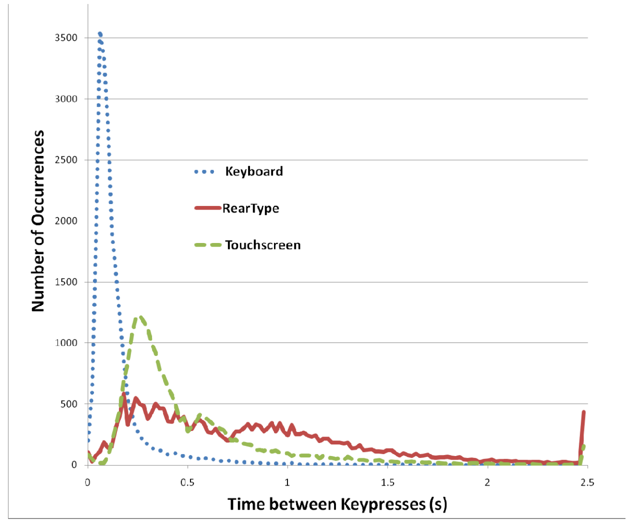


Figure 7. Histogram of time between consecutive keypresses for the three input methods, for advancing keypresses (i.e. at most one error in a row) in the experimental phase.

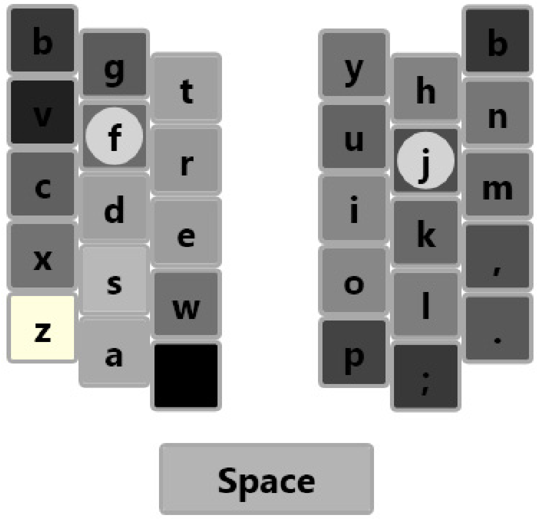
5.5 Key-By-Key Analysis

Figure 5 compares the use of backspace between the three input methods. Since our study software limited errors to a single character, the number of backspaces used is an appropriate measure of the number of errors. The mean number of backspaces typed was 28.5 across the three conditions. A repeated-measures ANOVA found a significant main effect for input technique on number of backspaces typed ($F_{(2, 22)} = 20.8$ $p<0.001$). Here the mean number of backspaces typed during an experimental task was 14.9 for keyboard, 44.8 for RearType, and 25.9 for the touchscreen. Pair-wise comparisons showed significant differences between all pairs of conditions ($p < 0.01$), with RearType performing worse than both keyboard and touchscreen.

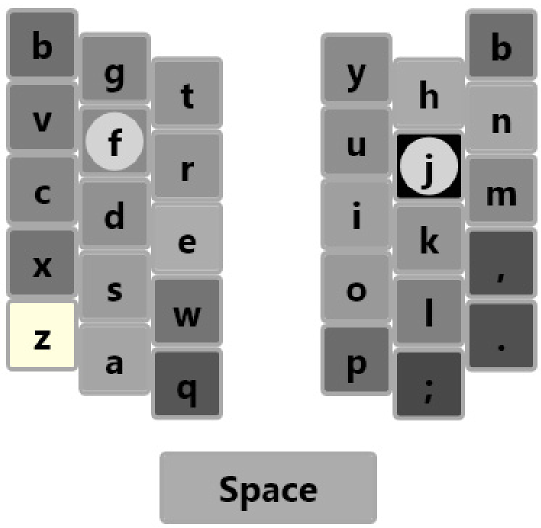
To help understand the sources of errors further, Figure 6 shows the confusion matrix for keys expected versus keys typed, for RearType and for a normal keyboard for comparison. RearType clearly had more errors, and the errors are predominantly horizontal off-by-one errors (the cells adjacent to the diagonal) and vertical off-by-one errors (the fainter diagonals on either side of the main diagonal). This is encouraging in combination with the typing speed achieved – it indicates that much of the time users have a working knowledge of where the keys are, after only an hour's exposure to the technique, but they need to improve their ability to accurately target.

Figure 7 illustrates the interval between consecutive keypresses for the three input techniques, and we can see that in addition to incorrect keypresses being deleted, participants were also slower in general at hitting keys with RearType, though there is a very large variation. (N.B. the Multi Function Panel hardware/driver appears to add a quantized latency.)

Exploring keypress delay further, Figure 8 shows delay on a per-key basis (the key being the one required, not necessarily the one typed). We can see that the users take longer when needing to press a key in the corners of the RearType keyboard than in the middle. This may be due to the difficulty of reaching the further keys – indeed P7 noted "*Difficult to type letter y because my*



(a) RearType



(b) Keyboard

Figure 8. Mean delay before a key is pressed for each non-shifted character. A black background represents the maximum for that input method, white is zero. Includes all “advancing” keystrokes (i.e. at most one error in a row). NB ‘z’ was not present in the experimental texts.

fingers are short”, and P12 stated “*Keys on the back of device may be too spread out for my hands to reach comfortably*”. We also observed two participants (P4, P5) holding their thumbs on the sides of the device, increasing their finger reach around the rear.

One simple solution to this is to reduce the size and depth of the keys slightly; the current prototype keys have the size and depth of travel of standard keyboard keys rather than the slightly smaller and shallower keys typically found on notebook PCs. The grip design could also be improved so as to allow users more mobility with their fingers, by allowing the user to pivot around the lower palm or even the wrist. P8 suggested that the “front” buttons could instead be on the sides, thus allowing more of the hand to be towards the rear of the device.

Another alternative would be to angle the top and bottom row keys so that reaching them requires less movement for the fingers. Taking this approach to its extreme, a group of keys surrounding each finger as for the DataHand Pro II might be used [4]. Yet another possibility is to use Green et al.’s reduced QWERTY keyboard [6], though this requires that multiple key strokes per character (KSPC) [13] are used in contrast to the prototype’s KSPC of 1.

5.5.1 Space and Backspace

Seven out of twelve participants (P1-2, P5, P8, P9-10, P12) noted in the post-survey questionnaire that they mixed up space and backspace sometimes, and would prefer that they were under opposite thumbs rather than the same thumb. The fact that they were under the same (right) thumb was a result of our initial prototype having modifier keys (Shift, Mode, Control, Alt) under the left thumb, though we removed them for the study as noted previously. Further exploration is required to ascertain the best distribution of thumb buttons.

Another cause of confusion with backspace is that it is normally found under a finger rather than the thumb, so this change is in contrast with QWERTY. One possibility would be to put backspace on the rear of the device in or near its normal position, but while this may be more intuitive, it also runs into the issues discussed previously with regards to the reachability of the corner keys.

6. CONCLUSION AND FUTURE WORK

We have presented the design and implementation of RearType, allowing text to be input using the back of a device, and therefore enabling the front to be relatively free from physical keys, on-screen touch keyboards, and the user’s hands occluding the screen.

Through a preliminary study with 12 participants who were expert QWERTY typists, we showed that with just one hour of training their typing speed for English prose on RearType averaged 15.1 WPM, which was as expected slower than their performance on a regular keyboard, but encouragingly was not statistically different from their performance with a touchscreen soft keyboard. Furthermore, users’ post-study responses indicated that they saw RearType as a good alternative to touchscreens. For expert users, RearType offers 10-finger input in comparison to hunt-and-peck using touchscreens or small thumb-operated keyboards.

Through our prototyping and study we have learned many useful lessons for future RearType-based devices. Three rows of five keys per hand worked well, though the corner keys were slightly harder to reach for some people. It is worth exploring smaller (notebook keyboard size) keys and changes in the grip so as to allow more freedom of movement of the fingers. Some participants found that simultaneously gripping and typing was uncomfortable, so other ways in which some of the weight can be supported are worth exploring (e.g. using the wrist or the top edge of the hand). Thumb keys are easily confused, so careful placement and design of the thumb keys must be used to avoid this. A visualization should be provided, but it should be easy to switch on and off manually (since it sometimes hinders rather than helps), and/or it should adaptively appear (e.g. when the Mode modifier is pressed).

This initial study of RearType addressed a few important questions, but leaves much to be done. One area is in running longer-term studies of learning spread across multiple sessions. This is required to map out the learning curve further and quantify expert input speeds. Longitudinal studies of use would address such issues as the ease of learning "Moded" keys and whether RearType can be useful for tasks other than typing prose. Methods for phasing out the visualization gradually as users become more expert must be further developed and tested empirically, so that RearType's potential for increasing usable screen real estate is realized.

We have highlighted many potential ergonomic improvements in grip, key size and positioning, device shape, and so on, that would be interesting to explore, both to improve user comfort and to minimize the effects of RearType's limitations on hand movement compared to keyboards. We have also discussed hardware and software innovations in the visualization to allow users to more easily learn RearType, e.g., using LucidTouch-style hover sensing so that users know what key their fingers are resting on. Finally, integrating RearType into a working Tablet, UMPC or e-ink device would undoubtedly involve interesting hardware and software challenges.

7. ACKNOWLEDGEMENTS

Thanks to Nicolas Villar for work on initial prototypes and testing, and Xiang Cao and Martijn ten Bhomer for pilot testing.

8. REFERENCES

[1] Baudisch, P. and Chu, G. Back-of-device interaction allows creating very small touch devices. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems* ACM (2009) 1923-1932.

[2] Clawson, J., Lyons, K., Starner, T., and Clarkson, E. The Impacts of Limited Visual Feedback on Mobile Text Entry. In *Proceedings of the International Symposium on Wearable Computing*, 2005.

[3] Conrad, R., & Longman, D. Standard typewriter versus chord keyboard: An experimental comparison. Ergonomics. 8. (1965) 77-88.

[4] DataHand Systems, Inc. DataHand Pro II ergonomic keyboard. http://www.datahand.com/

[5] Gopher, D., & Koenig, W. Hands coordination in data entry with a two-hand chord typewriter.Technical Report CPL 83-3. Cognitive Psychology Laboratory, Dept. of Psychology, University of Illinois (1983).

[6] Green, N., Kruger, J., Faldu, C., and St. Amant, R. A reduced QWERTY keyboard for mobile text entry. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*. ACM (2004) 1429-1432

[7] Grippity Ltd. Grippity 1.0 see-through keyboard. http://www.grippity.com/products.html

[8] Hinckley, K. and Sinclair, M. Touch-sensing input devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* ACM (1999) 223-230

[9] Hiraoka, S., Miyamaoto, I, Tomimatsu, K. Behind Touch: A Text Input Method for Mobile Phone by The Back and Tactile Sense Interface, In *Proceedings of Interaction 2003* IPSJ (2003) 131-138

[10] Kristensson, P.O. and Denby, L.C. 2009. Text entry performance of state of the art unconstrained handwriting recognition: a longitudinal user study. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM (2009) 567-570.

[11] Landsberger, H. A. *Hawthorne Revisited*, Ithaca (1958).

[12] Lyons, K., Starner, T., Plaisted, D., Fusia, J., Lyons, A., Drew, A., and Looney, E. W. Twiddler typing: one-handed chording text entry for mobile phones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* ACM (2004) 671-678

[13] MacKenzie, I. S. KSPC (keystrokes per character) as a characteristic of text entry techniques. *Proceedings of the Fourth International Symposium on Human Computer Interaction with Mobile Devices*, Springer-Verlag (2002) 195-210.

[14] Matias, E., MacKenzie, I. S., & Buxton, W. Half-QWERTY: A one-handed keyboard facilitating skill transfer from QWERTY. In *Proceedings of the INTERCHI '93 Conference on Human Factors in Computing Systems* ACM (1993) 88-94.

[15] SafeType, Inc. SafeType Ergonomic Keyboard. http://www.safetype.com/

[16] Sellen, A., Kurtenbach, G., & Buxton, W. The prevention of mode errors through sensory feedback. *Human Computer Interaction*. 7(2). (1992) 141-164.

[17] Silfverberg, M. Using mobile keypads with limited visual feedback: Implications to handheld and wearable devices.In *Proceedings of Mobile HCI 2003*. Springer-Verlag (2003) 76-90.

[18] Soukoreff, W., & MacKenzie, I.S. Text entry for mobile computing: Models and methods, theory and practice. Human-Computer Interaction, 17 (2002) 147-198.

[19] Sugimoto, M. and Hiroki, K. HybridTouch: an intuitive manipulation technique for PDAs using their front and rear surfaces. In *Proceedings of the 8th Conference on Human-Computer interaction with Mobile Devices and Services*. ACM (2006) 137-140.

[20] Vogel, D., Cudmore, M., Casiez, G., Balakrishnan, R., and Keliher, L. Hand occlusion with tablet-sized direct pen input. Proceedings of CHI 2009 – the ACM Conference on Human Factors in Computing Systems. ACM (2009) 557-566.

[21] Wigdor, D., Forlines, C., Baudisch, P., Barnwell, J., Shen, C. LucidTouch: A See-Through Mobile Device. In *Proceedings of the 20th Annual ACM Symposium on User Iinterface Software and Technology*, ACM (2007) 269–278.

[22] Wobbrock, J. O., Chau, D. H., and Myers, B. A. An alternative to push, press, and tap-tap-tap: gesturing on an isometric joystick for mobile phone text entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* ACM (2007) 667-676.

[23] Wobbrock, J. O., Myers, B. A., Aung, H. H. The performance of hand postures in front- and back-of-device interaction for mobile computing. International Journal of Human Computer Studies 12 (2008), 857-875.