

From Plastic to Pixels: In Search of Touch-Typing Touchscreen Keyboards

Leah Findlater

University of Maryland, College Park | leahkf@umd.edu

Jacob O. Wobbrock

University of Washington | wobbrock@uw.edu

Touchscreen devices have exploded onto the commercial stage in the past decade, most prolifically in smartphones, but in other forms as well, including tablets and interactive tabletops. While touchscreen devices have enormous appeal, one drawback is clear to anyone who has entered more than a few characters on one: Typing is slow, uncomfortable, and inaccurate, and it generally pales in comparison to typing on physical keyboards. A touchscreen's flat, glassy surface means that even expert typists have to look down at their fingers instead of feeling for the home row keys to situate their hands. Adding to the challenge: Whereas physical keyboards offer three input states—a finger can be completely off a key, resting on but not depressing a key, or depressing a key—touchscreen keyboards offer only the first and third of these states. Without a separate signal for pressure, touching is pressing, which makes crafting usable and pleasing touch-typing keyboards for touchscreens a significant challenge.

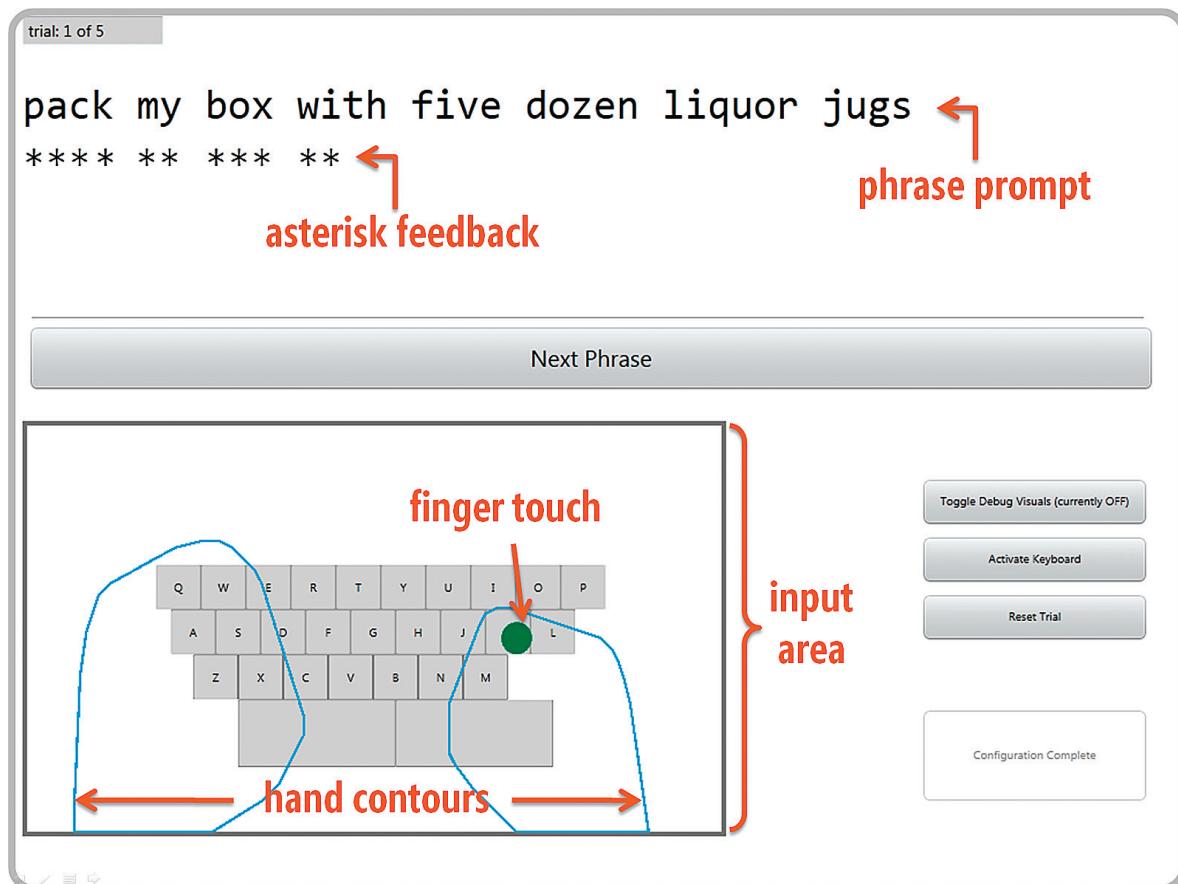
The upside is that touchscreens also offer distinct advantages. First, because touchscreen keyboards are

software based, they can support customization and adaptation of the keyboard in ways no physical keyboard can. Imagine, for example, you place your hands on an interactive tabletop and see the keyboard appear under your hands with the keys laid out to fit your unique typing style. Maybe you like to keep a comfortable amount of space between your hands or you find it difficult to reach the "Q" key with your left pinky—no problem, the keyboard will adjust. And being made from code and pixels means that personalized touchscreen keyboards can follow you to whatever devices you use.

Second, while most touchscreen keyboards use straightforward finger taps to enter text, touchscreens are also drawing surfaces and therefore offer gestural and multitouch capabilities that can be leveraged to great effect. Users are resistant to expending time to learn new text-input techniques—consider how few people use the Dvorak keyboard layout—but gestural text input has achieved some adoption through techniques based on the QWERTY keyboard (e.g., Swype

[www.swype.com] and SHARK2 [1], whose commercial name is ShapeWriter). In what other ways can touchscreen keyboards take advantage of these capabilities to complement the striking of keys?

Our research aims to improve the design of touchscreen keyboards by exploring these potential advantages. To this end, we have conducted a series of investigations on 10-finger touchscreen typing. In the first study [2] we examined how users type on a horizontal surface without any specific keyboard layout to conform to. This situation mimics touch typing with limited visual attention, so examining these typing patterns provides insight for how to redesign touchscreen keyboards to reduce visual attention. The second phase of our investigation [3] explicitly evaluated two types of adaptive keyboards that change over time to support an individual user's typing patterns. In the third phase of the work [4], we introduced multitouch gestures that can be used to complement keyboards by producing punctuation and symbols. Ultimately, our goal is to support fast and accurate touch



► Figure 1. Task interface, showing the asterisk feedback, visible keyboard condition (3). Hand contours and finger touch point are for illustration only and were not displayed to users.

typing with limited visual attention on touchscreens by employing a combination of improved design, personalization, and gestures.

Uncovering Natural Typing Patterns on “Flat Glass”

In this first phase of our research, we conducted a study with 20 expert typists to examine the patterns that emerged when users were asked to type on a flat surface, in some cases with no visual keyboard and no feedback.

By emulating touch typing with limited visual attention, we explored questions such as: When users are given no visual constraints about keyboard layout, where do they naturally place their fingers to strike each key? Do the outcomes resemble the usage of a



► Figure 2. Input area in the unrestricted (1) and asterisk feedback, no keyboard (2) conditions. Participants placed their thumbs over the red dots before typing in the otherwise blank area.

rectangular QWERTY keyboard? Will users have trouble hitting some keys consistently compared to other keys? Do typing patterns differ much from one user to the next?

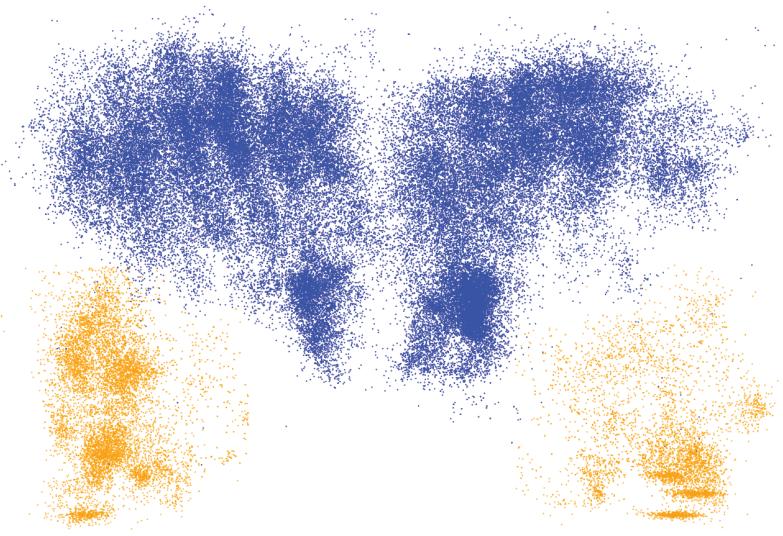
Mimicking an ideal keyboard.

Participants entered text on a

Microsoft Surface tabletop computer under three conditions (see Figures 1 and 2):

1. no feedback and no visible keyboard (unrestricted typing),
2. asterisk feedback and no visible keyboard, and

► Figure 3. Finger (blue) and non-finger (orange) touches in the *unrestricted* typing condition (1), showing space between hands, separate left and right spacebar areas, and evidence of forearms and heels of the hands resting on the screen. ($N = 20$)



3. asterisk feedback and a visible keyboard.

The two conditions without a visible keyboard (1 and 2) were designed to capture natural typing patterns. In the unrestricted condition (1), participants were unaware of spurious or missing touches, which mimicked an ideal touch-typing keyboard and allowed for the most natural typing possible.

In the *asterisk feedback* conditions (conditions 2 and 3), output for each non-space key press was in the form of an asterisk (*), similar to what one sees when entering a password (see Figure 1). The asterisk feedback provided the user with some indication that their input had been received without the system having to actually determine what letter the user had intended to press—a serious challenge when no keyboard was shown and participants could type wherever they liked! Participants corrected any typing errors they felt they had made with a backspace right-to-left swipe gesture so the asterisks and spaces lined up with the presented text. This requirement enabled a one-to-one mapping during analysis between touch events and letters from the presented text.

Redesigning the QWERTY keyboard.
In the unrestricted typing condi-

tion (1), in which participants could assume their input was correct without having to worry about accidental or incorrect key presses, typing speeds were on average 59 words per minute (WPM). While this speed was slower than the 85 WPM we saw when we gave these same participants physical keyboards, it was still quite fast for touchscreen text input.

Figure 3 shows a visualization of the typing data from this condition, combined across all users. Non-finger touches (orange) occurred frequently, about once per word, but were clearly separate from the finger touches. There was also a discernible space between the hands and an overall arched shape to the pattern of touches for both the left and right hands. In other words, these natural typing patterns did not look much like a standard rectangular keyboard.

Similar to a physical keyboard, participants often rested their fingers on the screen. However, once a phrase, sentence, or passage was under way, their fingers tended to lift until the desired text was complete. This insight should motivate designs that allow users to rest their hands without ill effects between typing episodes.

Figure 4 shows a visualization of typing patterns for the aster-

isk feedback conditions (2 and 3). Here, we were able to do a more detailed analysis of typing patterns by using the one-to-one correspondence between touch events and letters in the presented text. Since participants were asked to correct all errors they felt themselves make, typing speeds were slower (at 27 to 28 WPM) than in the unrestricted condition.

Unsurprisingly, as can be seen in Figure 4, participants were more consistent in where they placed their fingers when a visible keyboard was shown than when they were asked to type on a blank screen. Regardless of the keyboard condition, participants were also less consistent in where they touched their fingers for keys at the outer edges of the keyboard (Q, A, Z, P) than for keys in the middle. This pattern suggests that increasing the relative size of those outer-edge keys may improve typing accuracy.

These findings provide insight into how we might redesign QWERTY keyboards to better support natural typing patterns on touchscreens. Beyond these general implications, however, we also observed that typing patterns varied greatly from one user to the next, particularly in the no-keyboard conditions (1 and 2). Personalization may therefore be critical in supporting touch typing with limited visual attention on touchscreens.

Personalizing Touchscreen Keyboards

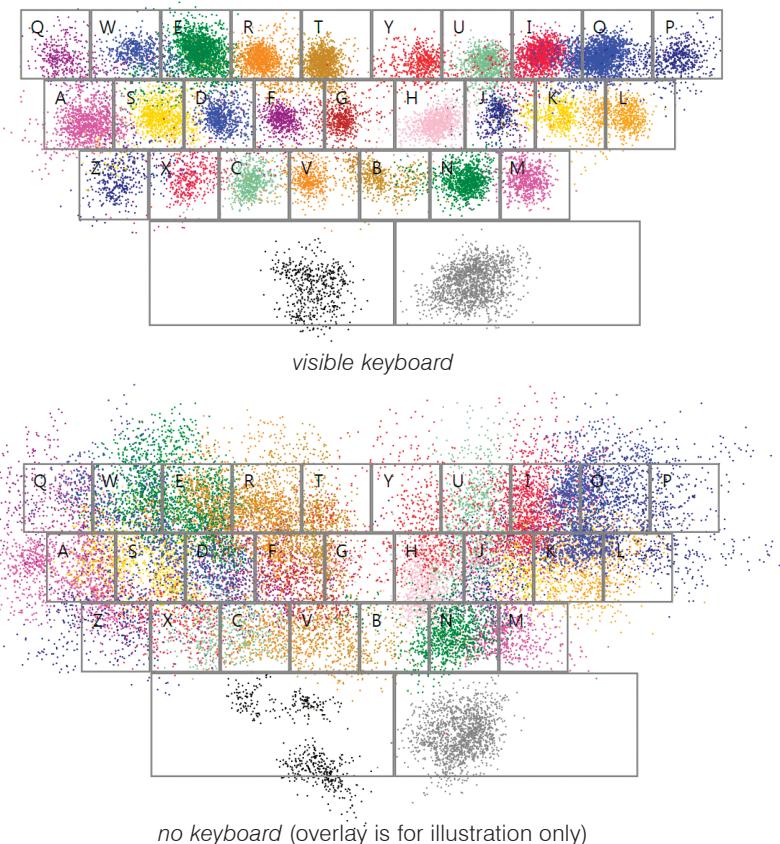
Based on our exploratory study, we found the idea of an adaptive, personalized keyboard to be compelling. In this second phase of the research, we designed and evaluated two novel adaptive keyboards for a Microsoft Surface. Both keyboards begin as a standard rectangular layout (Figure 5a) but over

time adapt to an individual user's typing patterns by personalizing how finger touches are mapped to specific keys. For the underlying key-press classification model, we used a C4.5 decision tree classifier (Weka's J48 implementation, provided by the Weka data mining toolkit: <http://www.cs.waikato.ac.nz/ml/weka/>), trained on finger location and movement features from each key press.

The two keyboards differed in one essential way. Independent of the underlying key-press model's adaptations, one keyboard always maintained a visually stable rectangular layout (Figure 5a). In comparison, the other keyboard visually changed to reflect the current state of the underlying model (Figures 5b and 5c).

We conducted a study with 12 participants to evaluate our personalized keyboards alongside a conventional rectangular QWERTY keyboard layout. Participants came in for three separate sessions and used each keyboard in each session. Again, as with our earlier study, participants were expert touch typists on physical keyboards.

Results showed that personalization can improve typing speed without injuring accuracy. By the third session, the personalized keyboard that did not visually adapt improved typing speed over the conventional keyboard by 15 percent (26.9 WPM vs. 31.0 WPM). This improvement in speed was accompanied by no detectable difference in error rates, which were less than 0.5 percent. Interestingly, although participants perceived this keyboard to offer good performance, they had trouble identifying the difference between it and the conventional keyboard, since both keyboards looked the same. (We did not tell participants during the study



► Figure 4. All key presses in asterisk feedback conditions, colored by key label. The visible keyboard shows more consistency across users than with no keyboard. ($N = 20$)

which keyboards were adapting and which were not.)

In comparison, the keyboard that did visually adapt provided no performance benefit over the conventional keyboard. The visually adaptive keyboard was perceived as comfortable and natural to type on, but some participants remarked that its unusual layout seemed to require more visual attention than the conventional static keyboard. This increase in visual attention is our hypothesized reason for the lack of a performance gain with this keyboard.

Thus, although personalization is not enough to bring touchscreen keyboards into parity with their physical counterparts, a well-designed personalized keyboard can be useful for improving performance. However, simply mimicking physical keyboards is not the only way in which touchscreen key-

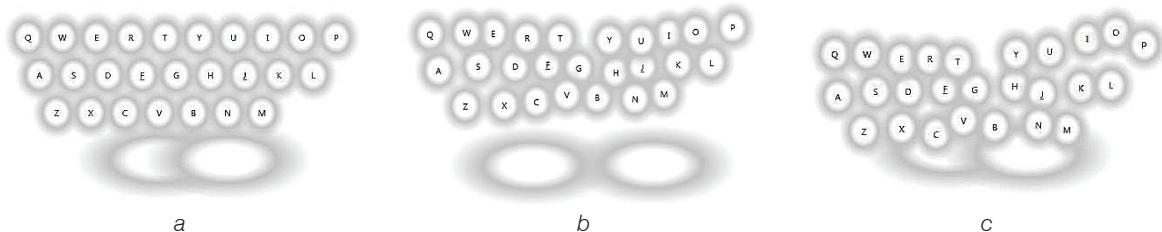
boards can be successful. Another way is by taking advantage of touchscreens for things physical keyboards cannot support, such as drawing gestures for punctuation and symbols.

Going Beyond QWERTY for Touchscreen Keyboards

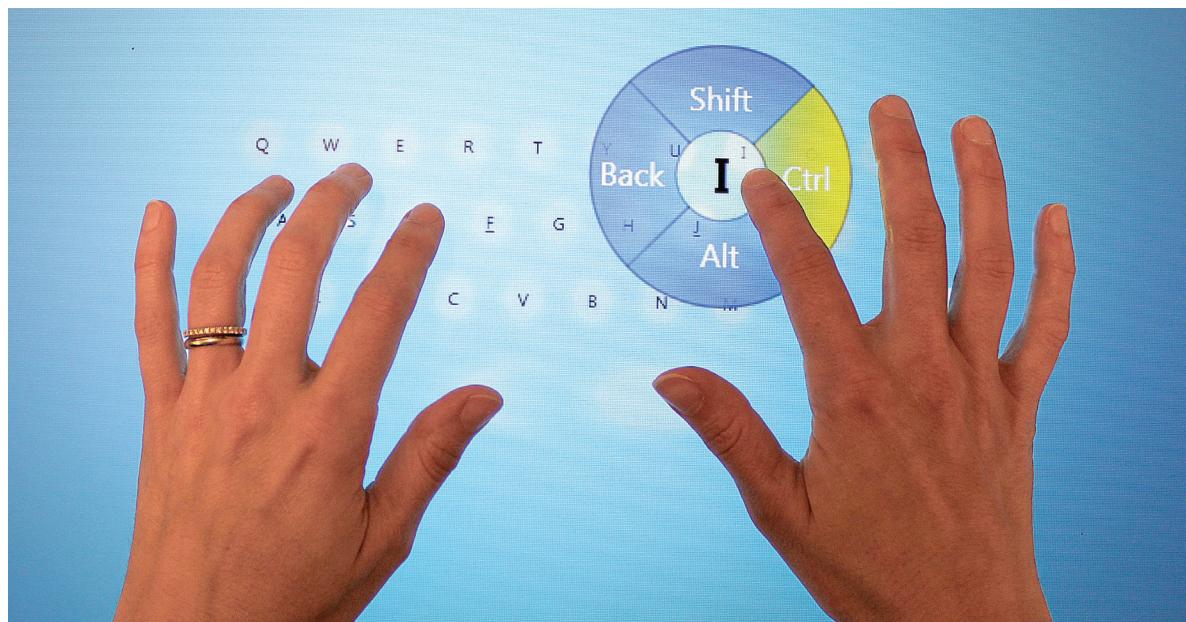
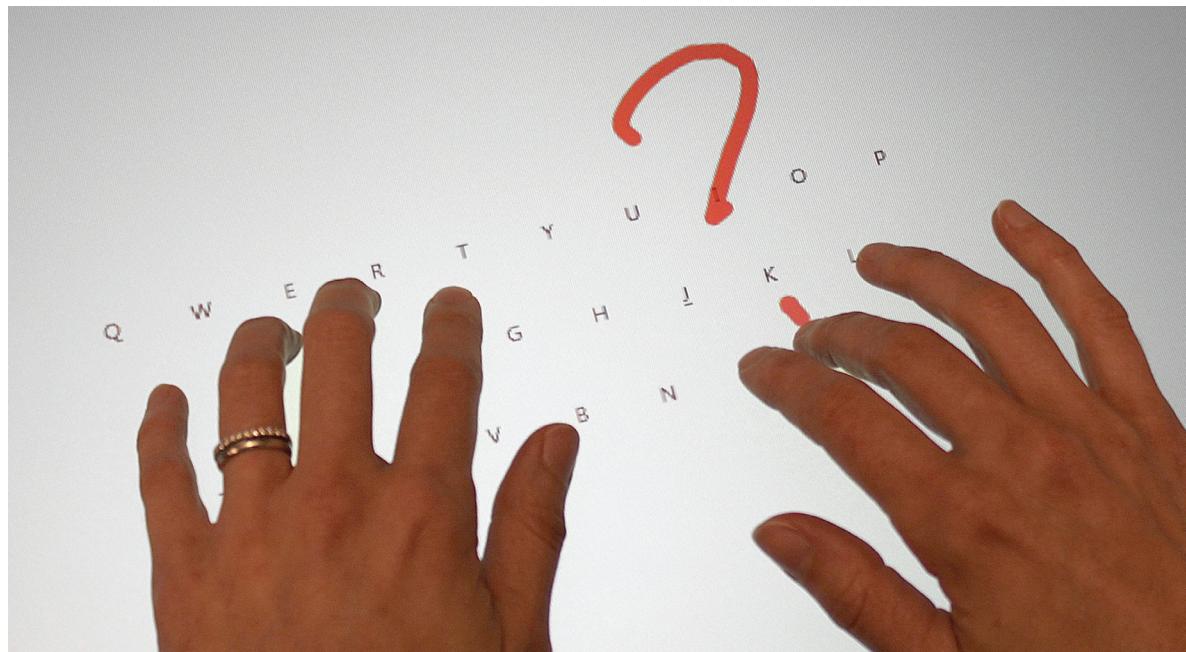
A limitation of previous research on touchscreen typing, including ours described here, is that researchers usually only consider the letter keys on the QWERTY keyboard and focus on faithfully replicating physical keyboards. To enter other symbols beyond letters, widely adopted commercial touchscreen interfaces require mode-switching with shift keys to alternate character sets. But touchscreens can support much more than keys: They can also support stroke gestures.

Our approach is to augment existing 10-finger QWERTY keyboards

► Figure 5.
Personalization with a visually stable layout (a), and two visually adaptive layouts from different participants in our study (b and c).

*a**b**c*

► Figure 6. (a) Our multitouch gesture technique, showing the end of the "z" gesture with all lefthand fingers down and one right-hand finger down; (b) A pie menu for modifier keys triggered by dwelling on the "I" key, making Ctrl+I the result in this example.



with multitouch gestural input, which exists as a complement to using mode-toggles, such as shift. With our multitouch gestural approach, shown in Figure 6a, users place four or more fingers down with their right or left hand and draw atop the keyboard with their other hand. Once all fingers are lifted from the screen, the drawn symbol is entered. This bimanual interaction uses active rather than passive modes, supports input with low visual attention, and does not require users to move their hands out of typing position. For a smaller device with two-finger or two-thumb input, this four-plus finger trigger could be replaced by pressing and holding with a single finger or thumb.

To create a set of guessable, intuitive gestures for non-alphanumeric input, we conducted a study to elicit user-defined gestures [5] from 20 participants. We asked them to create gestures, using however many fingers or hands they wished, for a set of 22 punctuation symbols (e.g., : #) and four common commands (space, shift, backspace, enter). The final set includes mostly single-touch gestures (requiring only one finger), with an additional multitouch option for four of the symbols (# : =). We also include multiple single-touch options for seven symbols, such as drawing “*” with three or four strokes or drawing “%” by connecting none, one, or both of the circles to the diagonal line on the same stroke. For commands, there was less agreement on what constituted a good gesture. As a result, we suggest that commands and modifier keys be provided through alternative mechanisms, such as keys on the primary keyboard or the pie menu shown in Figure 6b.

Can Pixels Ever Outperform Plastic?

As touchscreen devices are increasingly adopted and able to support a wide range of tasks, we need to devise ways of making our transition to such platforms successful. The entry of text is as fundamental to computing today as it ever has been. Although speech recognition continues to improve, it is not well suited to many types of text entry and usage environments. All too often we see users passing around or plugging in physical keyboards for text input instead of using the very touchscreen devices into which their text is going!

Our research provides several potential improvements to reduce the need for visual attention when typing on flat surfaces, taking advantage of the personalizable, adaptable, and gestural capabilities of these devices. Redesigning keyboard layouts to better support natural typing patterns and introducing personalized input models can improve typing performance. Novel gestural techniques can augment existing keyboards for stroke-based non-alphanumeric input. Our focus has been on 10-finger typing, but many of our findings should extend to smaller mobile devices as well.

Of course, language models are well understood, and existing approaches to combine touch input with language-model probabilities (e.g., [6]) should further improve typing performance. Localized vibrotactile feedback, perhaps even simulating the click of a key [7], may also further improve touchscreen keyboards.

Text entry has been part of nearly every computing system since the earliest days, and touchscreens—from small to large—will continue to be found in devices for many years to come. Although text entry and touchscreens were not conceived

in the same breath, their marriage is already inevitable. Our hope is to make this marriage a happy one by taking advantage of the unique characteristics of touchscreens, using personalization and gestures to make efficient keyboards from pixels, rather than from plastic.

ENDNOTES:

1. Kristensson P. and Zhai, S. SHARK2: A large vocabulary shorthand writing system for pen-based computers. *Proc. 17th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, 2004, 43-52.
2. Findlater, L., Wobbrock, J.O., and Wigdor, D. Typing on flat glass: Examining ten-finger expert typing patterns on touch surfaces. *Proc. SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, 2011, 2453-2462.
3. Findlater, L., and Wobbrock, J.O. Personalized input: Improving ten-finger touchscreen typing through automatic adaptation. *Proc. SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, 2012. To appear.
4. Findlater, L., Lee, B.Q., and Wobbrock, J.O. Beyond QWERTY: Augmenting touch-screen keyboards with multi-touch gestures for non-alphanumeric input. *Proc. SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, 2012. To appear.
5. Wobbrock, J.O., Morris, M.R., and Wilson, A.D. User-defined gestures for surface computing. *Proc. SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, 2009, 1083-1092.
6. Goodman, J., Venolia, G., Steury, K., and Parker, C. Language modeling for soft keyboards. *Proc. IUI '02*. ACM, New York, 2002, 194-195.
7. Poupyrev, I., Maruyama, S., and Rekimoto, J. Ambient touch: Designing tactile interfaces for handheld devices. *Proc. 15th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, 2002, 51-60.



ABOUT THE AUTHORS

Leah Findlater is an assistant professor in the College of Information Studies at the University of Maryland, College Park. Her research interests include personalization, accessibility, and information and communication technologies for development (ICTD).



Jacob O. Wobbrock is an associate professor in the Information School and an adjunct associate professor in computer science and engineering at the University of Washington. His research in HCI investigates novel user interface technologies, input and interaction techniques, human performance modeling, and accessible, mobile, and surface computing interfaces.