Hover Cursor: Improving Touchscreen Acquisition of Small Targets with Hover-enabled Pre-selection

Anna Ostberg
Synaptics, Inc.
San Jose, CA, USA
anna.ostberg@synaptics.com

Nada Matic
Synaptics, Inc.
San Jose, CA, USA
nada@synaptics.com

CHI'15 Extended Abstracts, Apr 18–23, 2015, Seoul, Republic of Korea.
ACM 978-1-4503-3146-3/15/04.
http://dx.doi.org/10.1145/2702613.2732903

Abstract
Even with highly-sensitive touchscreens and emphasis on "designing for touch", small target selection remains difficult. Good touch performance cannot solve the "fat-finger" problem, which results from occlusion and the size disparity between fingers and targets. We propose Hover Cursor, a method to improve small target selection using hover-sensing over a touchscreen. Using a capacitive touch sensor that also provides hover data, the hover position of the user's finger is displayed with a cursor, and selection is performed with a tap. In a Fitts' study, we compared Hover Cursor with direct-touch selection. Users made fewer selection errors with Hover Cursor. Hover Cursor was slower overall, but faster and more accurate for small targets.

Author Keywords
Touchscreens; proximity sensing; precise target acquisition; interaction techniques; hover.

ACM Classification Keywords
H.5.2. Information interfaces and presentation (e.g., HCI): User Interfaces. – Input devices and strategies, Interaction styles

Introduction
Direct-touch selection on mobile devices has become the dominant input technique. However, due to the

Figure 1. Twitter for Android is an example of an application that has small targets (to reply, retweet or favorite a tweet). Hover Cursor shows the position of the user's hovering finger as a blue semi-transparent circle that is 2mm in diameter. To reduce occlusion we applied vertical and horizontal offsets.

"fat-finger" problem, small target acquisition remains difficult and frustrating on touchscreen devices.

Many applications and websites use larger targets to address this, but this wastes valuable screen real estate. With high-resolution displays, a large amount of content can be viewed at a legible size. However, interactive elements are limited by physical size, not resolution, and must be large and touch-friendly. When targets are too small, users sometimes zoom-in to make them large enough to select. This requires an extra gesture and disrupts a user's context. Also, many mobile applications and websites do not allow zooming.

Some of the latest mobile devices feature hover-sensing (or proximity-sensing), using a capacitive touch sensor that also tracks finger hover [15]. This feature is likely to be in many more devices in the near future. This capability enables a pre-selection "tracking state", one of the basic input states described by Buxton [4], which is absent from existing touchscreen interactions.

We use hover-sensing to introduce Hover Cursor, a pre-selection method for touchscreen interfaces that eases small target selection. Our contribution is a systematic target selection technique that does not conflict with existing interactions, preserves the user's visual context, does not require additional hardware, and can be used with one hand. Additionally, Hover Cursor does not need to know the location of targets.

In an optimal implementation, the direct nature of touchscreen selection would be preserved, with Hover Cursor only used as necessary. In such a hybrid technique, normal selection would occur exactly as users expect: directly tapping a target. For difficult targets, a user will typically dwell over the target and a timeout can be used to invoke the Hover Cursor state.

In this paper, our goal is to find which situations, if any, are helped by the Hover Cursor interaction. Future work will explore the usability of the hybrid technique.

Related Work

Prior work falls into two major categories: small-target acquisition and hover input. For target acquisition, we focus on touchscreen input, without added hardware.

Small-Target Acquisition

Potter et al. [12] use offset cursor feedback and "take-off" selection to mitigate finger occlusion and allow users to adjust their position before selection. Albinsson and Zhai [1] propose arrow keys around the cursor or a leverage metaphor to provide pixel-level selection. Roudaut et al. [14] present techniques for thumb selection during one-handed use, by magnifying a region or using a cursor that is attracted by targets.

Several papers address finger input on interfaces that are designed for pen. In Shift [17], touching the display creates a callout of the occluded area and allows position adjustment before selection. This conflicts with existing touch gestures (such as panning) and depends on knowing target size for optimal performance. Escape [19] combines positional information with a gesture to select small densely-packed targets. The target's design indicates what gesture to use, limiting the visual design of targets. Additionally, the system must know the location of targets, an important limitation. Research by Benko et al. [3] takes advantage of multi-touch sensing. The most successful method uses a secondary finger to scale-up a region around the

Figure 2. The 2D-Fitts' task has a home location (currently active, red) and a target (currently inactive, white) for each trial. In (a) the user's finger controls the blue Hover Cursor approaching the home location. In (b), the Hover Cursor is in position over the home location. When the user taps, they will select the home location.

primary finger. This cannot be used with one hand and requires zoom, which disrupts a user's visual context.

Hover Input

Hover for input has been explored in the context of different devices. ThickPad [6] explores new interactions for hover in a notebook computer. Moeller and Kerne [11] describe a precision hover tracking system that can be applied to any display or area, but do not suggest specific uses. Hilliges et al. [8] use a camera for hover input over interactive tabletops.

Cheung et al. [5] use proximity sensing to enable traditional mouse hover interactions on touch surfaces. In Z-touch by Takeoka et al. [16], proximity sensing is used to draw and zoom. Unfortunately neither of these papers include testing or evaluation. Yang et al. [18] use cameras to detect approaching fingers and enlarge targets, improving selection performance. However, their solution depends on the system to be aware of target locations, modify target appearances, and requires additional hardware.

Hover Cursor Interaction

Hover Cursor provides continuous feedback for a hovering finger to ease target selection. The position of the user's hovering finger is displayed as a blue semi-transparent circle that is 2mm in diameter, see Figure 1. When a user positions the cursor over a target and taps their finger, the selection occurs at the location of the cursor, not the tap. A similar approach performed well in research by Pyryeskin et al. [13].

Hover Cursor uses the built-in hover-sensing capability developed by Synaptics [15] for the Samsung Galaxy S4 phone. This capability extends the capacitive sensing of the touchscreen to process finger hover. Touch-sensing tracks *x* and *y* position, while hover-sensing tracks *x*, *y*, and *z* (the height above the screen). When *z* is equal to zero, hover becomes touch. We perform basic filtering of hover data to mitigate finger jitter and potential sensor noise. Filtering also helps avoid positional drift as the finger taps to select.

To reduce occlusion, we apply two offsets based on experimentation in a prior pilot study. The vertical offset is 200 pixels (~11.5 mm) but shrinks smoothly near the bottom of the phone to enable selection of targets at the bottom of the display. There is a horizontal offset of 50 pixels (~2.9 mm), based on the handedness of the user. Handedness could be determined from the direction and trajectory of the hovering finger. However, in order to simplify the implementation and experiment, handedness was selected by the user before the study session.

User Study

We compared Hover Cursor to standard tap-only selection using a within-subjects design. The purpose of the study was to evaluate the speed and accuracy of target selection, with emphasis on small targets.

Task & Procedure

We used an internal testing tool implemented in an Android app [10] which includes a 2D-Fitts' task [7][9]. Our Fitts' task had eight indices of difficulty (*ID*), a function of target distance (*D*) and target width (*W*) defined as:

$$ID = \log_2\left(\frac{D}{W} + 1\right)$$

Based on ID, movement time (*MT*) can be predicted as:

$$MT = a + b(ID)$$

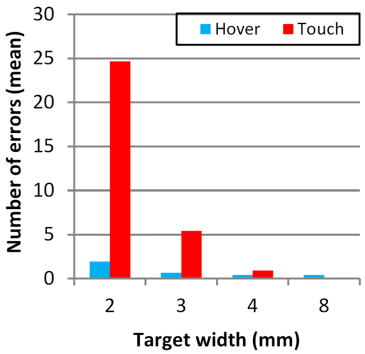

Figure 3. Total errors by width (mm), mean across participants.

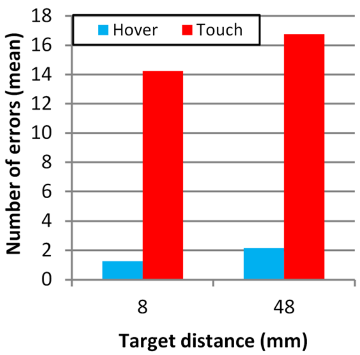

Figure 4. Total errors by distance (mm), mean across participants.

where *a* and *b* depend on the input technique. The test software (see Figure 2) varied target width (2, 3, 4, 8 mm) and distance (8 and 48 mm) to generate the eight ID's. Targets were presented in random order in each condition. Inactive targets were white, active targets were red. Target pairs were at angles of 45°, 135°, 225° and 315°. The test consisted of 32 targets per condition (4 widths x 2 distances x 4 angles). The software logged movement time and selection errors.

The study had two order-balanced experimental conditions: standard touch selection and Hover Cursor selection. Participants had unlimited practice time before each condition and were encouraged to balance speed and accuracy in the task. Upon tap, the next target pair appeared on screen, even if the user missed the target. For missed targets, an error sound played. Missed target pairs were repeated at the end of the task, to ensure one successful trial for each target pair. After each condition, a 5-point Likert scale was used to rate the interaction on several criteria.

Apparatus

The study was conducted on a Samsung Galaxy S4 (GT-I9500) smartphone with the AirView hover feature enabled. The Galaxy S4 has a 5" diagonal 1920-by-1080 pixel display. It can sense hover to about 20mm above the display. The phone used in the study was running Android Jellybean version 4.2.2.

Participants

Twelve people (7 male, 5 female) were recruited internally. All participants had experience with touchscreen phones. One participant was left-handed. Participants were seated for the task, with the phone resting on a table in front of them.

Results

We performed repeated measures ANOVA on our results.

Error Rate

An error is counted whenever the user misses a target. To capture error rate as a percentage, we consider the first 32 targets, before repetition of missed targets. On average, users had a 26% error rate for touch, and a 9% error rate for Hover Cursor. For the full task, including repetition of missed targets, users have significantly more errors with touch than with Hover Cursor ($F_{1,11} = 14.56$, $p = .003$). There is an *ID x device* interaction ($F_{7,16} = 3.50$, $p = .018$). There is a significant effect for width ($F_{3,20} = 42.57$, $p < .0001$), but no significant effect of distance ($F_{1,22} = .43$, $p = .52$) (see Figures 3 and 4). Least squares means testing shows a significant difference between touch and Hover Cursor for target widths of 2 and 3 mm. This reinforces the usability problems of small-target acquisition in existing touchscreen interactions.

Movement Time

Movement time (*MT*) is the time between selecting the home location and the target. On error-free trials touch is faster than hover, even for small targets (Figure 5).

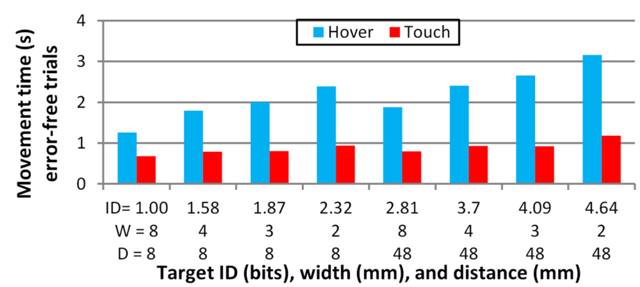

Figure 5. Task time in error-free trials by ID, width and distance.

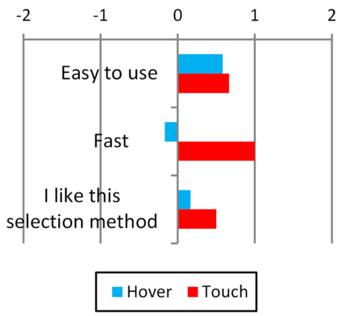

Figure 7. Mean response for positive questions (higher score is better).

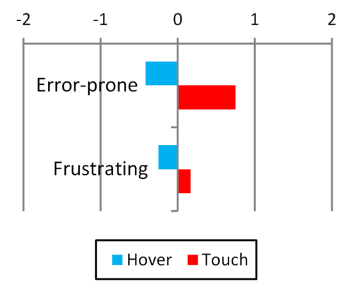

Figure 8. Mean response for negative questions (lower score is better).

However, when selecting small targets, errors are a common and frustrating occurrence. In real use, people need to make a selection, and if the target is small, they will have to persistently try to select it. We include all attempts on a target to reflect such a real-use situation. For total movement time, touch is slightly faster, but not by a significant difference ($F_{1,11}$ = 3.06, p = .11). There is a significant effect of *ID* ($F_{7,16}$ = 18.06, p < .0001) and *ID x device* interaction ($F_{7,16}$ = 6.4, p = .001).

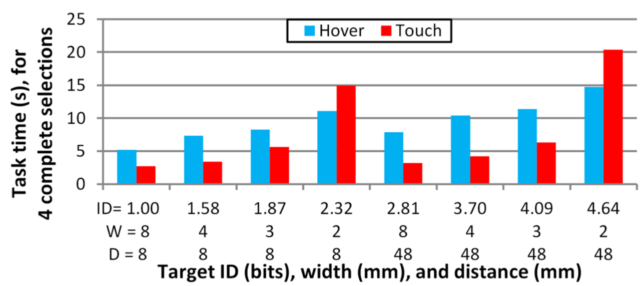

Figure 6. Total task completion time to get four successful selections, by ID, width, and distance (includes errors).

Figure 6 shows the extent to which Hover Cursor can help users select very small targets, particularly those that are 2 mm or less. The total time required to select such small targets is high in the touch condition.

Subjective Evaluation

After each condition, participants provided ratings on 5-point Likert scales for both positive and negative questions (see Figures 7 and 8). A score of '-2' corresponds to strong disagreement, and a score of '2' to strong agreement. Overall, ratings are mostly positive for both devices. We use a paired-samples t-test to assess significant differences. There is a significant difference between the perceived speed of touch and Hover Cursor (t(11) = 3.19, p = .008, two-tailed). Additionally, there is a significant difference for the error-prone question, with Hover Cursor rated less error-prone than touch (t(11) = 3.02, p = .01, two-tailed). The other questions do not have significantly different ratings.

Discussion and Conclusions

Hover Cursor provides a promising solution to the real-world problem of small-target selection. Even though we compare a novel concept with a well-established interaction, our solution fares quite well. It dramatically reduces errors for small targets, and does not increase them for large targets. Even though it is slower for larger targets, it reduces overall selection time for small targets. It is also encouraging that all the participants learned how to use Hover Cursor quickly and with minimal guidance. Subjective responses to Hover Cursor are not uniformly positive but indicate that users perceive it to be less error-prone than direct touch.

Our contribution addresses small-target selection on touchscreens and can be applied generically to any device with hover-sensing, with no additional hardware. Hover Cursor does not depend on two-handed use or require the system to know the locations of targets. It does not require changes to targets or changes to the user's context through zooming or magnification.

Some participants commented that they wanted to tap on large targets, but still have access to Hover Cursor for small targets. This fits with the hybrid technique discussed earlier. This would most likely be determined by a timeout, similar to the method used in Shift [17] and the hesitation gesture guides in Medusa [2]. We intend to implement this hybrid solution and evaluate it using similar methods. Such a solution may provide an effective balance of speed and accuracy for selecting targets of all sizes on touchscreens.

References

[1] Albinsson, P. and Zhai, S. High precision touch screen interaction. *CHI '03*,ACM Press (2003), 105-112.

[2] Annett, M., Grossman, T., Wigdor, D., and Fitzmaurice, G. Medusa: a proximity-aware multi-touch tabletop. *UIST '11*, ACM Press (2011), 337-346.

[3] Benko, H., Wilson A., Baudisch, P. Precise selection techniques for multi-touch screens. *CHI '06*, ACM Press (2006), 1263-1272.

[4] Buxton, W. A three-state model of graphical input. In *INTERACT '90*, North-Holland Publishing Co. (1990) 449-456.

[5] Cheung, V., Heydekorn, J., Scott, S., Dachselt, R. Revisiting hovering: Interaction guides for interactive surfaces. *ITS '12*, ACM Press (2012) 355-358.

[6] Choi, S., Han, J., Kim, S., Heo, S., and Lee, G. ThickPad: a hover-tracking touchpad for a laptop. *UIST '11*, ACM Press (2011), 15-16.

[7] Fitts, P. The information capacity of the human motor system in controlling the amplitude of movement. Journal of Experimental Psychology 47 (1954), 381-391.

[8] Hilliges, O., Izadi, S., Wilson, A., Hodges, S., Garcia-Mendoza, A., and Butz, A. Interactions in the air: adding further depth to interactive tabletops. *UIST '09*, ACM Press (2009), 139-148.

[9] ISO, ISO 9241-411 Ergonomics of human-system interaction – Part 411: Evaluation methods for the design of physical input devices. International Organization for Standardization, 2012.

[10] Mockler, J. UX suite: a touch sensor evaluation platform. *MobileHCI '14*, ACM Press (2014), 631-636.

[11] Moeller, J. and Kerne, A. ZeroTouch: an optical multi-touch and free-air interaction architecture. *CHI '12*, ACM Press (2012), 2165-2174.

[12] Potter, R.L., Weldon, L.J., Shneiderman, B. Improving the accuracy of touchscreens: an experimental evaluation of three strategies. *CHI '88*, ACM Press (1988), 27-32.

[13] Pyryeskin, D., Hancock, M., Hoey, J. Comparing elicited gestures to designer-created gestures for selection above a multitouch surface. *ITS '12*,ACM Press(2012),1-10.

[14] Roudaut, A., Huot, S., Lecolinet, E. TapTap and MagStick: Improving one-handed target acquisition on small touch-screens. *AVI '08*, ACM Press (2008), 146-153.

[15] Synaptics ClearPad® Touch Technology Powers Samsung Galaxy Note 3. http://goo.gl/XaYO4Q

[16] Takeoka, Y., Miyaki, T., Rekimoto, J. Z-touch: an infrastructure for 3D gesture interaction in the proximity of tabletop surfaces. *ITS '10*, ACM Press (2010), 91-94.

[17] Vogel, D. and Baudisch, P. Shift: a technique for operating pen-based interfaces using touch. *CHI '07*, ACM Press (2007), 657-666.

[18] Yang, X., Grossman, T., Irani, P., and Fitzmaurice, G. TouchCuts and TouchZoom: Enhanced Target Selection for Touch Displays using Finger Proximity Sensing. *CHI '11*, ACM Press (2011), 2585-2594.

[19] Yatani, K., Partridge, K., Bern, M., Newman, M. Escape: a target acquisition technique using visually-cued gestures. *CHI '08*, ACM Press (2008), 285-294.