

Experimental Analysis of Touch-Screen Gesture Designs in Mobile Environments

Andrew Bragdon¹, Eugene Nelson¹, Yang Li² and Ken Hinckley³

¹Brown University
Providence, RI, USA

{acb, ebnelson}@cs.brown.edu

²Google Research
Mountain View, CA, USA

yangli@acm.org

³Microsoft Research
Redmond, WA, USA

kenh@microsoft.com

ABSTRACT

Direct-touch interaction on mobile phones revolves around screens that compete for visual attention with users' real-world tasks and activities. This paper investigates the impact of these *situational impairments* on touch-screen interaction. We probe several design factors for touch-screen gestures, under various levels of environmental demands on attention, in comparison to the status-quo approach of soft buttons. We find that in the presence of environmental distractions, gestures can offer significant performance gains and reduced attentional load, while performing as well as soft buttons when the user's attention is focused on the phone. In fact, the speed and accuracy of bezel gestures did not appear to be significantly affected by environment, and some gestures could be articulated eyes-free, with one hand. Bezel-initiated gestures offered the fastest performance, and mark-based gestures were the most accurate. Bezel-initiated marks therefore may offer a promising approach for mobile touch-screen interaction that is less demanding of the user's attention.

Author Keywords

Touch, gestures, mobile phones, performance, evaluation

ACM Classification Keywords

H5.2 Information interfaces and Presentation: Interaction.

General Terms

Human Factors

INTRODUCTION

Touch screens are growing rapidly in popularity as an input method for smart phones and other mobile devices [1]. These devices have few, if any, hard buttons; instead soft buttons are the dominant command invocation paradigm on commercial touch-based phones, and are effective when the user is seated and focusing directly on the phone, with no distractions (an *ideal environment*).

However, users rely on mobile devices while sitting, walking, driving, and in diverse environments with various distraction levels [2]. When the user is in a *non-ideal environment*, such as walking through an airport looking for a currency exchange, she must navigate, maintain awareness of her location and avoid obstacles. These *situational impairments* [3] are environmental factors that inhibit the ability to perform tasks on a device. We hypothesize, therefore, that techniques such as soft buttons, which require the user to

look at and press a relatively small target, will suffer performance degradation in non-ideal environments, while also potentially degrading performance of other concurrent tasks. This implies that for some mobile tasks, soft buttons could be augmented with other techniques that demand less attention.

In this regard, direct touch gestures offer some possible advantages. Some direct-touch gestures have the potential to be articulated eyes-free or with reduced visual monitoring, making them more resilient to distraction. Gestures can be committed to muscle memory, which helps users focus on their task [4], and it is possible to articulate some gestures with one hand. Gestures also require no dedicated screen space, which is a limited resource on mobile phones.

Even if gestures can be performed quickly and accurately with reduced attentional load in mobile environments, reliance on visual feedback is still a problem. Audio may be sufficient for some tasks, even complex tasks such as scheduling meetings [5]. In other cases, feedback on a heads-up display may be sufficient, e.g. in a car, via a wireless network [6]. Some cases may require little feedback, such as a gesture that is mapped to call a specific person. In many situations, a quick glance may be all that is needed, e.g. when a gesture brings up a map of the current location. Thus, we believe there is significant value in reducing the attentional burden required by soft buttons.

Thus, the goal of this paper is to explore the design space of touch gestures on mobile devices, and evaluate these designs in multiple mobile environments that induce varying levels of situational impairment on users. We induce situational impairment as two factors: *motor activity* and *distraction level*. For motor activity, we examine sitting and walking. For distraction level, we examine no distraction, a light situational-awareness distraction, and an attention-saturating distraction.

We explore two factors of gesture design: *moding* and *gesture type*. Since touch often scrolls or pans by default, the user needs a robust method to enter gesture mode. We explore several ways to indicate mode, including crossing through the screen bezel to integrate mode selection with the articulation of the gesture itself. We also analyze two prevalent types of gestures: *mark* and *free-form path* gestures.

Thus this paper yields insights on a number of fundamental questions regarding touch gestures on mobile phones:

- Which performs best in common mobile environments with varying levels of motor activity and distraction: soft buttons or gestures?
- How is attentional load affected by gestures?
- Can gestures be made eyes-free with sufficient accuracy in the above environments?

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or to publish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.

Copyright 2011 ACM 978-1-4503-0267-8/11/05...\$5.00.

- Can gestures be articulated efficiently one-handed?
- Among the factors and conditions we consider here, what is the fastest combination of moding techniques and gesture type for mobile phones?
- To what extent do varying levels of distraction and motor activity affect the speed and accuracy of gestures?
- Which approach do users prefer? Would users like to have both soft buttons and gestures?

Our analyses of these questions contribute a fundamental body of knowledge about touch screen gestures on mobile phones that has not been adequately addressed by prior work.

RELATED WORK

Empirical Studies

A field study of two types of service workers concluded, as one of three design recommendations, that executing actions should not demand high visual attention [7].

Li et al. compared 5 mode switching techniques for performing pen-based gestures on Tablet PCs, and found that using the non-dominant hand to control a button was the fastest [8]. In contrast, we focus on smaller mobile phones, using touch gestures – which necessitates different techniques and experimental design appropriate for mobile devices.

Pirhonen et al. [9] compared a media player with five unmoded gestures to a player with five soft buttons, both worn by a user while walking. In addition, [9] asked users to wear the phone on the hip; while typical for media-players, we believe this is not representative of the common phone-usage scenarios in which the user holds a phone, which is our focus. From this inspiration, we extend this work by exploring moding type, gesture type, distraction type, and motor activity, and by examining task performance with finer granularity, attentional load and eye gaze.

Lee and Zhai evaluated the hard and soft button performance on mobile devices with multiple types of feedback and input methods [10]. A study of one-handed soft button use found users could select small targets reliably and that walking did not affect speed or accuracy [11]. Lin, et al. studied selecting targets with a stylus while sitting and in two mobile environments [12]. Kane et al. [13] found a significant interaction between walking/not walking and button size, and proposed larger buttons when users are walking. Our focus is not keyboard/keypad input, but rather command selection (e.g. back in a browser) which, as discussed below, is distinct.

Related Techniques and Systems

Bezel Swipe [14] is a technique for scrolling and multi-selection on mobile devices which defines sub-regions of the bezel for different command modes; the user can then cross these regions to enter a mode. Manual Deskterity [15] uses a similar approach to create objects. This work forms the basis of the bezel moding technique presented in this paper.

MacKay et al. compared techniques for scrolling on mobile devices and found differences in performance between sitting and walking [16]. BlindSight [5] lets users perform interactive tasks on phones eyes-free while on a call, using physical buttons and audio cues, while [17] showed that gestures can be learned faster than keyboard shortcuts on desktop PCs.

Techniques Not Evaluated

See Table 1 for a discussion of other moding techniques not evaluated in this paper.

Menus

Marking menus [4] help users move from novice to expert by showing a spatially arranged hierarchical menu that uses compass-aligned marks; the menu is shown on a delay so that after a time users can simply perform the marks from memory. This design inspires the mark-based gestures evaluated in this paper. Simple marking menus [18], in which users draw separate strokes (e.g. 3 strokes for up-right-up) were shown to be more accurate and slightly faster than single-stroke compound marks. However, this requires a “transient mode”: if the user has input 2 strokes and may input a third, the system must wait for some time threshold for a third mark. We were concerned this might tempt users to look at the screen to ensure their marks had registered as intended; we thus used single-stroke marks, although we note that accuracy could be improved with simple marks. A tilt-based marking menu [19] was explored on mobile phones, but task times were high – a 2-segment gesture required a mean of 3.9 – 4.7 s.

OVERVIEW

We examine four factors: moding technique, gesture type, user’s motor activity, and distraction level of the environment. Below we discuss each factor in more detail.

Karlson et al. [20] observed that users prefer to use smart phones one-handed – that is, holding and supporting the phone, as well as executing commands, with a single hand – and do so the majority of the time. This is not surprising since using both hands incurs the opportunity cost of the additional hand, and greater effort. Given these advantages, we chose to focus on the single-hand scenario, and so all our techniques are designed to work well with a single hand.

Learnability of gestures is also an issue; [17] found it outperformed keyboard shortcuts. Approachability is also a problem; it has been shown, however, that users of a GestureBar disclosure mechanism can use gestural UIs without prior training or experience [21]. Therefore we focus on expert performance, since such an approach could be adapted to disclose the gestures and moding techniques discussed here.

Technique	Evaluation Rationale
Pressure	Li et al. [8] explored pen pressure to initiate a mode switch, however, in the context of touch, increasing pressure also increases the force of friction between the user’s finger and the screen, which is not the case with pens, and may be uncomfortable. In addition, this variable is not available on standard capacitive touch screens.
Clickable screen	Some devices, such as the Blackberry Storm (www.blackberry.com), let the user bear down on the touch surface: exceeding a certain pressure threshold causes the surface to “click” in, much like a mouse button. The click is used for picking, while swiping without clicking is used for scrolling. This leaves dragging while the screen is clicked in, which has the same discomfort issue as pressure.
Press-and-hold	Li et al. [8] showed that press and hold with a 1-second hold time was the slowest of the mode switching techniques they tested, and also was rated the worst. This technique therefore did not seem promising for touch.
Multiple fingers	Lepinski et al. adapted marking menus for multi-touch [39], supplementing directional marks with 5-finger chording. Since chording cannot be done one-handed on the screen, we did not implement this approach.
Hard buttons	Not tested for direct command invocation, as touch-based phones have few, if any, hard buttons.
Voice recognition	Can be used to execute commands, however, cannot be used in many mobile environments due to social acceptability (e.g. a meeting, etc.).

Table 1. Techniques not evaluated.

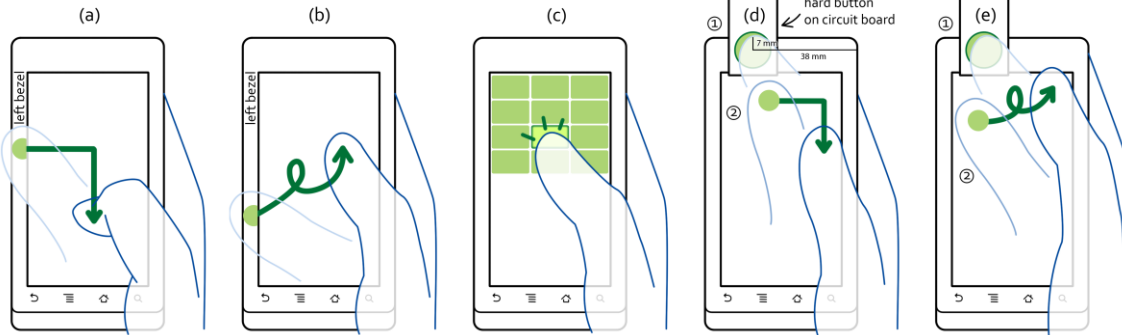


Figure 1.

Command invocation techniques: (a) bezel marks, (b) bezel paths, (c) soft buttons (control), (d) hard button marks, (e) hard button paths. Figures to scale.

MODING TECHNIQUES

This section summarizes the gesture-moding techniques explored. Vibrotactile/audio feedback was not provided, as selection of command buttons on phones such as the Android/iPhone does not give feedback; also consistent with [8].

Hard Button-initiated Gestures. Li et al. [8] showed that using the non-dominant hand to press a physical button while drawing pen gestures with the dominant hand was the fastest of five mode switching techniques tested. Since our study focuses on single-handed mobile devices use, we could not transfer this bimanual technique directly. In pilot testing with 2 users performing the main experiment (see below), we tested using a button on the side of the phone that could be pressed with the index finger while holding the phone. However, users found this awkward since they had to brace the phone against their palm to counteract the force of the button press; they also had a strong tendency to look down at the phone “to make sure they didn’t drop it.” We then tested switching modes by soft button; however, in pilot testing with 2 users they often missed the button, even after we made it quite large (20x20 mm); we also noted users often looked to ensure they hit the button.

Therefore, we instead mounted a physical button just above the top of the phone’s screen (Fig. 1d/e). The 14x14 mm button is made of a rubberized material that feels quite different from the glass phone bezel. The hard button requires force to depress, creating a mechanical “click.” Without looking, users can rest and feel their finger on the button before pressing, and be confident that they have clicked it. Once the button is clicked, users can draw a single-stroke gesture anywhere on the screen, at any scale. On contact-up, it is recognized. The user does not have to hold the button, but rather presses and releases it to enter gesture mode; the next stroke is treated as a gesture. In pilot testing, users were able to comfortably reach the button, one-handed.

Bezel Gestures. We extend [14] to act as a moding technique for path gestures: *bezel gestures* combine the tasks of setting gesture mode, and performing the gesture itself, into one fluid motion (Fig. 1-a/b). The user swipes through a bezel of the screen, setting gesture mode, continues by drawing a gesture, and finally releases contact to execute the command. In a sense, bezel gestures combine a crossing interface [22] that has large crossing targets with a gestural interface.

On contact up, the stroke input is fed to the gesture recognizer, and the appropriate action is registered. In our implementation on the Motorola DROID’s capacitive touch screen, if the beginning point of the input stroke was very close to the

edge of the screen, such an input was considered a bezel gesture, otherwise it was deemed normal input. This heuristic worked quite well in pilot testing, and is the same used in [14]. The tactile feedback of touching the bezel confirms to users that they are contacting the bezel without having to look. Bezel gestures include information on *which* of the 4 bezels was contacted (bezels were not subdivided into smaller targets), e.g. performing a gesture from the left bezel could connote a different action than from the right.

Soft Buttons. To offer a performance baseline, we implemented a soft-button condition (Fig. 1c), with 12 buttons separated by 1-mm gaps in a 3x4 grid (14.2x9.5 mm). The buttons are similar to and slightly larger than the numeric keypad soft buttons in the Android 2.1 Phone application (12.7x7.9 mm buttons, 2-mm gap). In comparison, the mean width of the adult male index finger and thumb is 18.2 mm and 22.9 mm; adult female, 15.5 mm and 19.1 mm [23]. Though smaller than an average finger, these buttons are typical in size of many smart phones, and we observed in pilot testing are easily pressed with the thumb, consistent with [24] [25] [26].

Each button is white with a black single-letter label. In most real-world toolbars and menus, commands are not listed in alphabetical order; mirroring this, we randomly chose 12 letters and assigned one to each button in random order (static). We chose this toolbar/menu layout rather than a numeric keypad to model how most commands are selected on mobile phones, and also to mirror the unfamiliarity in button layout by the unfamiliarity of specific gestures in the gestural techniques (see below). As an example, in the iPhone web browser, while unrelated other than both are browser functions, the Switch Tabs and Bookmarks button are adjacent.

GESTURE TYPES

We paired the moding techniques with 2 types of path-based gestures frequent in the literature: rectilinear, mark-based gestures and free-form gestures. We separate these two types here to better understand their performance characteristics; in a real application, both types could be used. Gesture type also affects what type of recognizer can be used, which in turn may affect performance. Although free-form path gestures are more complex than marks, our goal in testing was to identify design tradeoffs in a mobile context relative to the mark gestures, since they are used frequently in the literature. As with soft buttons (above), we used a gesture set of size 12.

Mark-based Gestures. Mark-based gestures, as exemplified in marking menus [27], are typically comprised of axis-aligned (up/down/left/right and optionally four diagonals, e.g. “north-east”) rectilinear mark segments that form a com-

pound path, e.g. “up” followed by “right” (Fig. 2, right). Mark-based gestures are simple and thus quick to execute and potentially tolerant of imprecision due to rapid execution.

To maximize recognition tolerance, we limit the mark-based gestures explored to only axis-aligned marks. Our gesture set is comprised of the 12 marks shown in Fig. 2 right. We implemented a simple recognizer that examined the starting/ending point and bounding box of the gesture, combined with simple thresholds and heuristics to identify the gestures.

Free-form Path Gestures. Free-form gestures naturally support 2-D operands (e.g. lasso select), can be rotationally invariant, and may be mnemonic (e.g. scribble delete [28]).

We implemented a set of 12 gestures (Fig. 2, left) that met the following criteria: relatively simple to perform, includes no axis-aligned marks, distinct enough from one another to be robustly recognized while allowing for imprecise drawing. We started with 9 gestures from past systems and studies (Fig. 2). We added 3 gestures representative of the complexity used in prior studies, such as [17], so that the gesture set as a whole was distinct as to be robustly recognized.

Recognition was implemented using the built-in template-based Android 2.1 OS recognizer (used by a variety of commercial applications), similar at a high level to [29].

GESTURE TYPE + MODING COMBINATIONS

Each of the moding techniques was crossed with the gesture types (see Table 2). Hard-button-initiated gestures could be drawn anywhere. Bezel path gestures could be drawn from any bezel. Bezel mark gestures had to start from a specific bezel corresponding to the direction of the mark’s first segment (e.g., N-W would start from the bottom bezel).

MOBILE ENVIRONMENT

We model mobile environment as two factors.

Motor Activity

Karlson et al. [20] studied three motor activities: sitting, standing and walking. We eliminated standing as in pilot testing with 2 users performing the experiment (see below) we observed no major change in performance from sitting. We felt using a mobile device while running is rarer than these other activities, and so eliminated it.

Sitting. Perhaps one of the most common motor activities while using a mobile device is sitting: e.g., at the office, in a meeting, on a bus, while driving, etc. Users sat upright in a non-reclining chair adjusted to be 0.45 m high (Fig. 3).

Walking. To test walking in a controlled way, we used a motorized treadmill in a room with only the experimenter present (Fig. 3). While previous work (e.g. [11]) has asked users to walk in a populated building, which is more ecologically valid, we opted for a more controlled approach. We wanted to capture distraction task performance as well as

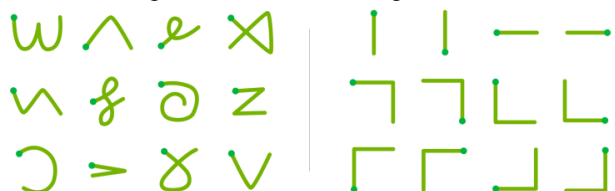


Figure 2. Free-form gestures (left) incl. pigtail [35], spiral [36], downward caret [17], Z [37], X [38], upward caret [38], “delete” [38], “squiggle” [17], and “redo” [28]. Mark-based gestures (right). Dots indicate start of each gesture.

Technique	Gesture Type + Moding Combination
Bezel Marks	Bezel Mode Switch + Mark-based Gestures
Bezel Paths	Bezel Mode Switch + Free-form Path Gestures
Hard Button Marks	Hard Button Mode Switch + Mark-based Gestures
Hard Button Paths	Hard Button Mode Switch + Free-form Path Gestures
Soft Buttons	See Above

Table 2. Command Invocation Techniques

eye-gaze data of the user looking at the phone, which would be difficult to do reliably in an uncontrolled environment. Knoblauch et al. recommend using walking speeds ranging from 0.91 m/s to 1.22 m/s for design purposes, depending on age and other factors [30]. We therefore chose 0.91 m/sec as the target speed. We measured and confirmed that the treadmill was operating at 0.93 m/s (S.D. = 0.014).

Distraction Level

We used 3 distraction levels, with varying attentional load.

No Distraction. This distraction level represents scenarios in which a user concentrates solely on operating the phone. Command-invocation tasks were displayed on the phone screen in the bottom-left corner. Thus, no eye movement is needed as the task and UI are displayed together.

Moderate Situational Awareness Task. This distraction level represents scenarios in which users must maintain some awareness of their environment, e.g. watch for a bus stop, follow a presentation/meeting, etc., but need not devote all of their attention to the task. We based the design of this task on successive sensory inspection sustained attention tasks [31].

To simulate such environments in a controlled way, a 17” monitor was set up so that the user could not see both the phone and screen simultaneously; a horizontal line was configured below which the user was required to hold the phone (Fig. 3). Phone tasks were displayed on one half of the monitor, the distractor task displayed on the other. A red circle appeared; users were asked to press and hold the spacebar of a keyboard with their non-dominant hand while the circle was visible. The circle randomly appeared for 1 to 8 whole seconds, then disappeared for 1 to 8 whole seconds. The relatively slow-changing but unpredictable nature of this task meant users spent most of their time watching the circle, but could glance down quickly at the phone with relative ease.

Attention-Saturating Task. We implemented an attention-saturating dual-task framework [31] in which users performed a continuous attention-saturating task (AST) while simultaneously performing additional tasks on the phone. The demand placed on attention by the phone tasks can then be measured as a drop in performance in the AST. Like the situational awareness task, the AST was shown on one half of the external monitor with command tasks on the other.

As in [32], the goal of the attention-saturating task was to keep a moving circle centered on a fixed crosshair. Random forces caused the circle to move in a simple 2D physics simulation in which it had momentum (Fig. 3). The user controlled the circle using an elastic tether connected to the mouse cursor, controlled via a desktop mouse. Raw perfor-

Environment	Motor Activity + Distraction Level Combination
Direct	Sitting + No Distraction
Indirect	Sitting + Situational Awareness Task
Walking	Walking + Situational Awareness Task
AST	Sitting + Attention-Saturating Task

Table 3. Environments

mance was measured as the circle's mean distance from the crosshair, sampled every 500 ms. For this task, users held the phone in their right hand and the mouse in their left so as to be consistent with the other conditions. This potential confounding factor is mitigated by the fact that performance on the AST task was normalized relative to baseline data (see below), all users were right-handed, and the AST task was performed using the left hand for all techniques.

Environment Combinations

We did not fully cross motor activity and distraction because we found that some combinations were not representative of common scenarios (see Table 3). Sitting+AST approximates an attention saturating experience, perhaps similar to driving; while talking or texting when driving is unsafe [33], users do issue commands on their phones while driving, e.g. to place calls, and we wanted to measure the extent to which this would affect attentional load (AST performance) and eye-gaze switches. We did not explore Walking+AST since a user would not likely want to use a phone during such a scenario. We also did not consider Walking+No Distraction as we believe that most users always maintain some level of environment awareness when walking and using a phone.

EXPERIMENT

Participants, Equipment and Physical Setup

We recruited 15 participants from the general population of Brown University (mean age 20.8, S.D. 2.73, 6 female, all right-handed). We advertised widely to get a sample of participants with diverse backgrounds and computers expertise; the average self-rated computer expertise was 5.06 on a 7-point Likert scale where 4 was "Intermediate" and 7 was "Expert"; 4 users reported using a touch device regularly.

We used Motorola DROID smart phones running the Android 2.1 update 1 OS. We disabled the built-in hard buttons (e.g. volume rocker switch, etc.) in software. Phones measured 60x115x13.7 mm, weighed 169 g, had a glass capacitive multi-touch screen (94 mm diagonal, 854x480 px), 256 MB RAM, and bezel widths of 4, 4, 10 and 8 mm on the left/right/top/bottom. The phone communicated with a laptop (visible only to the experimenter) via network, outputting to a 17" XGA monitor for displaying the primary and distractor tasks where appropriate. Participants held the phone in their right hand. We used 2 phones, one with the added gesture hard button and one unmodified (for all other techniques); the weight difference between the 2 phones was negligible, and the button assembly did not affect how users held the phone. Experimental software was Java-based, with one ap-

plication running on the phone and another on the laptop.

Eye gaze data was recorded using a video camera placed at a fixed distance recorded the user's face including eye movements (similar to [32]); the videos were analyzed offline to the frame level to determine eye movement start and stop times. The distance between phone and display was large enough to very reliably identify eye movements.

Tasks

Our goal in this study was to measure expert performance, and so we sought to simulate an environment in which the user had memorized the gestures in question. This approach is similar to [27], in which the users were asked to draw rectilinear marking menu gestures based on descriptions of the form "NE", which inherently indicate the sequence of marks required (in this case, a "north" mark followed by "east"). However, for our study, such acronyms do not include all necessary information, such as which bezel to start from, etc. Thus, we opted to use simple icons that disclose this information (for soft buttons, this showed the command label, e.g. 'Q'). For distraction-free tasks, the icon appeared in the bottom left corner of the phone's screen (icons were too small to trace but large enough to disclose the gesture), and for distraction tasks, the icon appeared on half the external monitor (Fig. 3, above). Feedback (correct/incorrect) was displayed on the screen (phone screen for direct, monitor otherwise) after each performance.

Experimental Design and Procedure

We used a repeated-measures within-participants experimental design. After a pre-questionnaire, participants were introduced to the study; read a description of the first command-invocation technique and shown a 40-second demonstration video. Users were given a short demonstration and then asked to complete a series of tasks using the technique in the first environment: 2 training blocks followed by 6 measured blocks; each block contained each of the 12 commands once. We felt 12 was a sufficient number of commands to be representative of a typical mobile application. For environments involving a distractor task, participants were given 3 minutes to become accustomed to the distractor task in isolation before using the technique concurrently. This was repeated as follows:

15 participants
 x 4 environments/distraction combinations
 x 5 command invocation techniques
 x (2+6) blocks (training + measured)
 x 12 commands
 = 28,800 trials completed

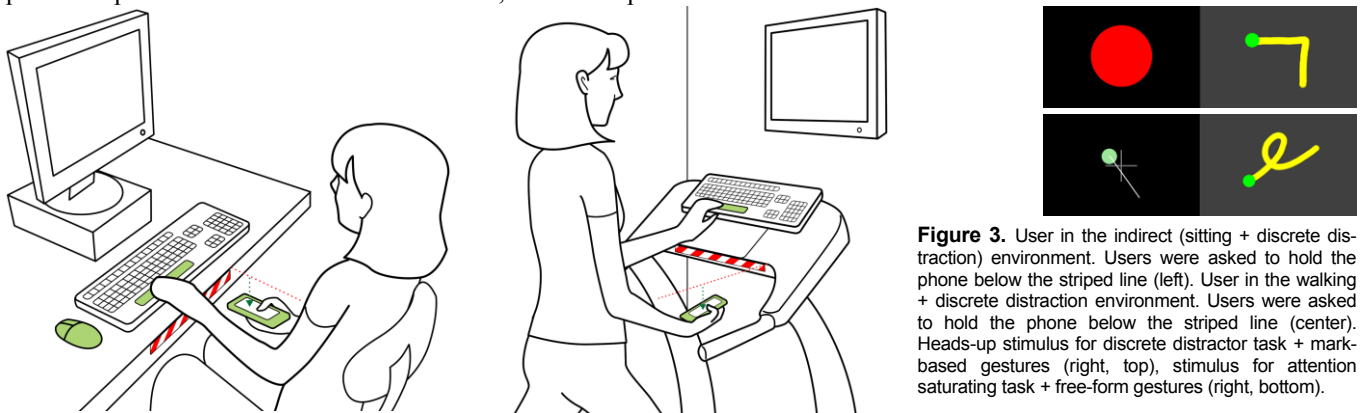


Figure 3. User in the indirect (sitting + discrete distraction) environment. Users were asked to hold the phone below the striped line (left). User in the walking + discrete distraction environment. Users were asked to hold the phone below the striped line (center). Heads-up stimulus for discrete distractor task + mark-based gestures (right, top), stimulus for attention saturating task + free-form gestures (right, bottom).

Condition order was counterbalanced with randomization. After completing 5 techniques and before moving on to the next environment users had a 7-minute break. Dependent variables included completion time, mode errors, command errors, and baseline and concurrent distractor task performance. Gesture performances were recorded. Cameras recorded users' eye movements in context.

Participants were paid \$15 per hour plus \$0.25 for every 100 ms faster than 2,000 ms they completed tasks on average (integer units) for a given technique/environment combination while maintaining less than a 10% error rate average (i.e., they earned an extra \$1 if they completed with an average of 1,600 ms with 94% accuracy). Users were notified when their average passed a 100-ms threshold and the accuracy was sufficiently high, with current average speed displayed in fractional seconds. We implemented this performance-based compensation system to keep participants motivated, after observing in pilot testing that some participants lost interest in the tasks partway through the experiment. Once complete, users filled-out a post-questionnaire.

RESULTS

Before analyzing variance, we observed histograms of the data and observed that it appeared to fit the normal distribution. Training blocks are excluded from this analysis.

Time

We report mean completion time results in Fig. 4, 5. Analysis of variance with technique as a within-subjects factor¹ found a significant main effect ($F_{2,67,37,33}=47.23, p<0.001$). There was a significant main effect for environment ($F_{3,42}=4.83, p<0.01$). There was no significant technique \times environment interaction ($F_{12,168}=1.31, p=0.22$). Post-hoc comparisons were performed using 2-tailed t-tests, with Holm's sequential Bonferroni adjustment [34] for multiple comparisons.

Overall Performance. We began by looking at mean completion time by technique averaged across environments. While this gives equal weight to each environment, which may not be representative of usage habits, it provides an overview of technique performance.

Bezel marks had the lowest mean completion time, 1092.87 ms; this was significantly lower than soft buttons ($t_{14}=5.12, p<0.0038$), which had a mean of 1244 ms (an improvement of 12.17%). There was no significant performance difference between soft buttons and hard button mark's mean of 1262 ms ($t_{14}=-0.46, p=0.66$). The 8.08% increase from soft buttons to bezel path's 1344 ms was not significant ($t_{14}=-3.11, p=0.0078$). Finally, there was a significant, 14.1% increase in mean completion time between bezel paths and hard button path's 1534 ms ($t_{14}=-5.68, p<0.0033$), the highest mean.

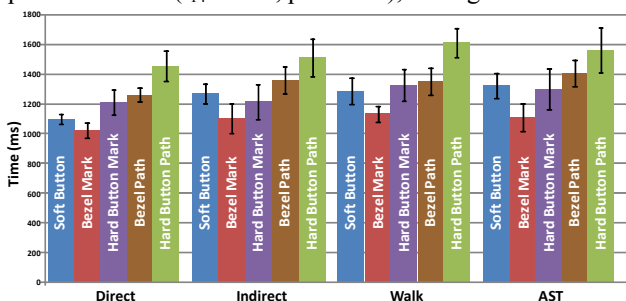


Figure 4. Task completion time by technique and environment.

Performance within Each Environment. To explore the surprising result that bezel marks outperformed soft buttons, we made additional comparisons focused on each of the 4 environments. In direct, without distractions, bezel marks had a mean completion time of 1022 ms, 6.82% lower than soft button's 1097 ms. There was no significant difference between bezel marks and soft buttons ($t_{14}=2.52, p=0.025$), surprising given that the user was not distracted and concentrated directly on the phone.

When used in the indirect environment, bezel marks (1102 ms) significantly outperformed soft buttons ($t_{14}=3.70, p < 0.0045$), which had a mean of 1271 ms. In the walk environment, bezel marks (1134 ms) significantly outperformed soft button's 1286 ms ($t_{14}=3.41, p < 0.005$), and in AST, bezel marks (1112 ms) again significantly outperformed soft button's 1321 ms ($t_{14}=3.84, p < 0.0042$).

It is notable that bezel marks and soft buttons performed similarly in direct, and that with various distraction types, bezel marks significantly outperformed soft buttons in each case.

Effect of Environment on Performance. We conducted further tests to see how the two best-performing techniques, bezel marks and soft buttons, were affected by environment. We also examine bezel paths (which we hypothesized might be affected by distractions).

For soft buttons, there was a significant difference in performance between the direct and indirect environments ($t_{14}=5.26, p<0.0036$) of 15.84%. Interestingly, there was no significant difference for soft buttons between indirect and walk or indirect and AST ($p>0.05$). This suggests that in terms of performance, the biggest effect for soft buttons came from not looking directly at the phone at all times.

For bezel marks, on the other hand, there was no significant difference in performance between direct and indirect, indirect and walk or indirect and AST ($p>0.05$), suggesting bezel marks were unaffected by not looking directly at the phone.

For bezel paths, completion times increased from direct to indirect by 7.85%; however, this difference was not significant ($t_{14}=2.12, p=0.052$). For walk and AST, bezel paths remained within 3.27% of indirect performance. This was surprising given that vibration or distractions could be expected to affect free-form gesture performance.

Accuracy Analysis

Mean accuracy and errors type results are reported in Fig. 6. Analysis of variance with technique as a within-subjects factor showed a significant effect for accuracy ($F_{4,56}=29.94, p<0.001$), as did environment ($F_{3,42}=5.09, p<0.01$). Technique \times environment¹, had no significant interaction effect

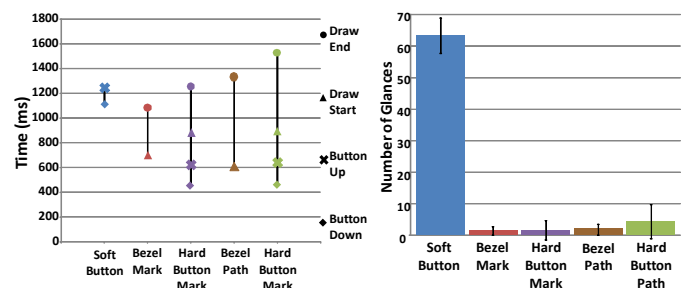


Figure 5. Task timing breakdown by technique (left), glances at phone in environments other than direct (right).

($F_{9,52,133,27}=0.50, p=0.88$). The same post-hoc multiple comparisons adjustment procedure was used as above.

Overall Accuracy. As above, to get a sense of the overall performance, we examine mean technique accuracy across environments (including both mode-in and command errors).

Hard button marks (93.5%) and bezel marks (92.5%) had the highest mean accuracy, followed closely by soft buttons (91.2%). There was no significant difference between hard button marks and bezel marks in accuracy, or between bezel marks and soft buttons ($p>0.05$). Bezel paths (81.4%) and hard button paths (83.5%) had the lowest mean accuracy. Hard button paths had significantly lower accuracy than soft buttons ($t_{14}=5.59, p<0.0036$).

Error-Type Analysis. Of the two moding approaches—bezel and hard button—there was 9.63% greater mode-in errors for hard buttons than for bezel; however, this was not significant ($t_{29}=0.19, p=0.85$). Of the two gesture approaches – mark (0.69 errors/block) and path (1.86) – there were significantly more (168%) command errors for paths than marks ($t_{29}=11.24, p<0.000001$).

Environmental Effect on Accuracy. From direct to indirect, soft button accuracy decreased by 2.88%, but this was not significant. Differences in soft button accuracy between indirect/walk and indirect/AST were not significant ($p>0.05$).

From direct to indirect for bezel marks, accuracy decreased by 2.93%, but was not significant ($t_{14}=1.90, p=0.078$). Differences in mean accuracy for bezel marks between indirect/walk and indirect/AST were less than 1.16%. Similarly, hard button marks performed consistently across environments; differences in accuracy between direct/indirect and between indirect/walk and AST, were less than 1.28%.

For bezel paths, from direct to indirect, accuracy decreased by 4.35%, but was not significant; from indirect to walk, the difference in accuracy was only 1.23%, and from indirect to AST the accuracy decreased by 3.45% but was not significant ($p>0.05$). For hard button paths, from direct to indirect,

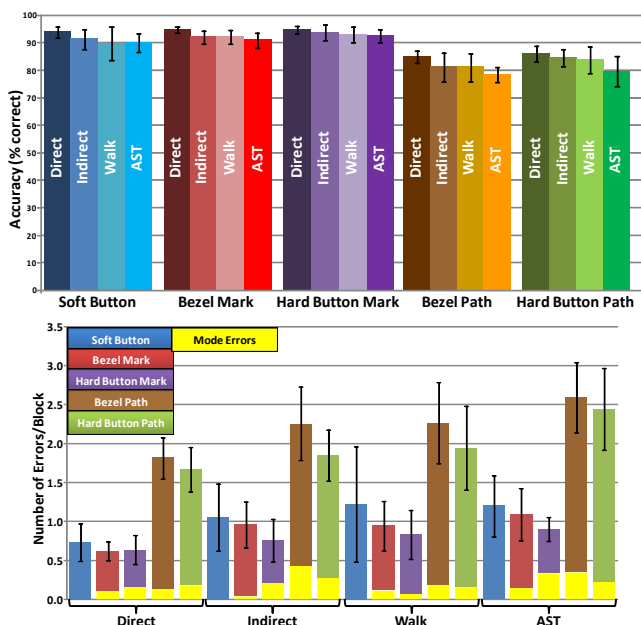


Figure 6. Accuracy rates by technique and environment (top); errors/block by technique and environment with mode errors shown (bottom).

and indirect to walk, the differences in accuracy were all less than 1.74%; from indirect to AST, however, the difference was 5.91%, but not significant ($t_{14}=1.68, p=0.11$).

Accuracy within Environments. Hard button marks, bezel marks, and soft buttons were the most accurate. For direct, accuracy differed by at most 0.8% and for indirect, walk and AST, at most 3.2%. No significant differences were found for environment ($p>0.05$).

The path gesture-based techniques were consistently the least accurate. For direct, hard button paths were significantly lower than soft buttons by 7.8% ($t_{14}=-4.89, p<0.0038$). This general trend continued, with hard button paths having 6.6% lower accuracy for indirect, 6.0% for walking, and 10.4% for AST. Very similar trends were seen for bezel paths.

Discrete Distractor Performance

For sitting, the four gesture-based techniques' mean discrete distractor performances (Fig. 7, right) were within 2.1% of each other. However, soft buttons performed 6.6% worse than bezel marks, a significant difference ($t_{14}=2.66, p<0.025$). For walking, a similar pattern held, with the gesture techniques having mean performances within 3.3% of one another. Soft button performance was 4.2% lower than bezel marks, but this was not significant ($t_{14}=1.81, p=0.093$).

Continuous Distractor Task Performance

To control for differing distractor task aptitudes, we normalized performance with the baseline mean performance from the second half of the distractor-only training (Fig. 7, left).

Soft buttons had the highest (worst) mean normalized distance from the target (2.1), while bezel marks had the lowest mean (1.6), significantly lower than soft buttons ($t_{14}=-2.61, p<0.025$) – Fig. 7, left. Hard button marks and bezel paths were within 9.7% of bezel marks. Finally, hard button paths had a mean distance of 1.9, but this was not significantly different from bezel marks ($t_{14}=2.11, p=0.053$). The relatively high normalized scores are likely caused by the continuous administration of command invocation tasks, which likely served as a greater distraction than occasional invocation.

Eye Gaze

The difference in the number of glances (Fig. 5) for soft buttons and all the gestural techniques was significant ($t_{14}=20.04, p<0.000001$). Interestingly, there was no significant difference for glances between any of the gestural techniques ($p>0.05$). The mean percentage of tasks requiring looks across environment for soft buttons was 98.8%; in contrast, the mean for gestural techniques was 3.5%. Since users were free to look, it is clear that looking was required for soft buttons but essentially not needed for gestural techniques. Looking was expensive, requiring 522 ms to finish looking down for soft buttons (plus 768 ms of looking at the phone).

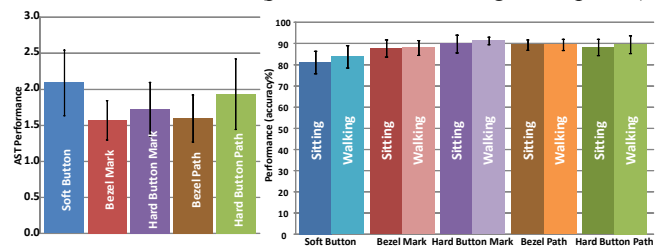


Figure 7. Normalized performance on the attention-saturating task (left), discrete distractor task performance (right).

Subjective Preferences

For direct, a majority (8 of 15 users) chose soft buttons as the preferred technique for performance; 4 chose hard button marks and 3 chose bezel marks. For *least* preferred, 10 chose hard button paths, with the rest evenly distributed.

For indirect, 0 now preferred soft buttons; 7 of 15 users chose bezel marks, 7 chose hard button marks, and 1 chose bezel paths. For their *least* preferred technique, 7 now chose soft buttons. 7 users wrote unprompted that a gesture-based technique allowed them to not have to look at the phone to perform commands: e.g. bezel marks “was still easy to use without looking at the phone”; and “I felt like I needed to look” for soft buttons. This trend continued for walking, again 0 users preferred soft buttons; 10 users chose bezel marks, 4 chose hard button marks, and 1 chose bezel paths. Finally, it further continued for AST, 9 users chose bezel marks and 6 chose hard button marks. Despite the vibration of walking, users appeared to still prefer gestural techniques; “[it is] easy to draw rectilinear shapes while not looking at the phone.”

For overall preferred technique, 7 chose bezel marks, 6 chose hard button marks, and 1 chose soft buttons. This is a surprising result given the ubiquity of soft buttons. When asked for their least favorite overall, free-form path gesture techniques dominated; 9 chose hard button paths and 4 chose bezel paths. Users commented on the relative ease of making rectilinear marks over free-form paths; one user wrote, “simple straight lines and right angle turns are easier to make with high tolerance of error.” 2 users felt pressing the hard button was an “extra step,” but this did not appear to be universal.

Keystroke Level Analysis

Fig. 8 shows a keystroke level analysis. Several general trends become apparent: first, it is notable that for the soft button task, although users finish looking down after 457-564 ms in indirect tasks, completion time increases by only 174-

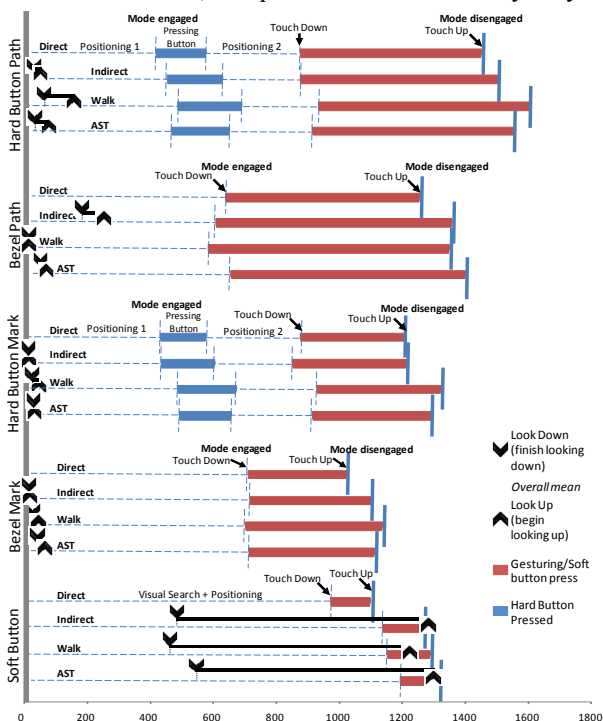


Figure 8. Keystroke-level analysis of gesture performances.

225 ms; we attribute this primarily to user reaction time. In addition, it is also notable that gesturing consistently begins sooner for bezel moding than hard-buttons. It is also notable that users were able to consistently begin pressing the hard button before the bezel mode became engaged; however, given the cost of then repositioning the hand, users completed the gestures after the bezel conditions.

Independent Analysis of Path Gesture Recognizer

The gesture error rates of the path gesture techniques were higher than 10%, so an independent, single-blind human analysis of the recorded performances was done to test if the recognizer malfunctioned. 5% of path gestures performed were randomly sampled and then human-recognized with a specific “recipe” set up *a priori*². Comparison of the software to the human recognition found 3.5% false positives and 3.3% false negatives. As even for humans 100% agreement is difficult to achieve, we feel this discrepancy is acceptable.

We attribute most of the path gesture errors made (Fig. 9) to user error: quickly drawing free-form path gestures on a phone with a thumb while holding the phone may be more difficult than, say, clicking buttons with a mouse, in which error rates are typically < 4%. Specifically, we hypothesize that it is the nature of the thumb that makes it difficult to accurately perform gestures one-handed – indeed, accuracy rates likely would be higher with two-handed input. We believe bezel marks were successful because the recognizer can be quite robust given the simple nature of the gestures. We believe also that had participants performed the gestures more slowly they would have had a lower error rate. We also note that the error rates for gestures were typically within 700 basis points of the soft button rates. It is possible that with a different type of recognizer, performance could be improved.

Stability of Performance over Multiple Blocks

There was a significant effect of block number on task completion time ($F_{5,1794}=2.87, p<0.05$), but not on error rate, however ($p>0.05$). The improvements were seen primarily for the gestural techniques. Soft buttons improved by 38 ms (3.6%), and actually worsened slightly in the last block (-0.97%). Bezel marks improved by 11.1%, bezel paths by 7.6%, and hard button marks improved by -0.4%. After extended training (perhaps several weeks), gesture performance could improve. We believe this threat to validity is mitigated by the fact that the learning effect for soft buttons appeared to be limited, and gestural learning did not appear to be of sufficient magnitude to indicate the performance measured was not representative.

It is important to discuss how representative user performance was. It is notable that we were looking at command invocation rather than data entry, with the soft buttons representing command buttons (e.g. “Create Contact,” etc. but not “1, 2, 3, ...”). We feel that for this context, it is less likely, for many users, to fully memorize the location of toolbar button/menu items, the way they would for say QWERTY, and instead may perform a visual search coupled with adjacency cues from memory. While there is an obvious benefit to a



Figure 9. Examples of users' path-based gesture command errors

user to memorize keyboard locations, we believe toolbar buttons are used less frequently and so there is less need to fully memorize them. Looking at the soft button learning effect, this was minimal during the experiment, as expected.

We note mean soft-button speed recorded for direct (1097 ms) was less than other studies, such as approximately 500 ms from [24]. We attribute the difference to differences in methodology, e.g. labeled targets, and non-reciprocal tasks.

DISCUSSION

Which performs best: soft buttons or gestures? Surprisingly, when users were looking directly at the phone there was no significant difference in completion time between soft buttons and bezel marks, probably because targets were not highlighted but needed to be identified from a letter caption. We believe this is realistic, however, since soft buttons for executing commands typically have either a text or icon label. In addition, bezel marks significantly outperformed soft buttons for the other three environments (sitting with discrete distraction, walking with discrete distraction, and sitting while doing an attention-saturating task). We hypothesize from the eye-gaze data that most of the performance benefit for gestures came from not having to look down at the phone.

How is attentional load affected by gestures? Performance on the discrete distractor task and the AST were both significantly improved for bezel marks over soft buttons. Hard button marks and bezel paths both performed similarly to bezel marks. This result is excellent, showing that gestures can reduce user’s attentional load over soft buttons, even though gestures involve reproducing geometry with some degree of accuracy and soft buttons can be performed in ~1.2 s.

Can gestures be made eyes-free in the above environments with sufficient accuracy? The eye-gaze data indicates that gestures can be made eyes-free, requiring looking at the phone just 3.5% of the time. In contrast, soft buttons required looking 98.8% of the time.

Can gestures be done one-handed effectively? Mark-based gestures can be done one-handed effectively (e.g. no significant difference in accuracy for bezel marks and soft buttons); however, our analysis indicates that accurate one-handed free-form path gestures may be considerably more difficult.

What is the fastest tested combination of moding technique and gesture type? Most accurate? Bezel marks appear to be the fastest combination of moding techniques and gesture type, followed by hard button marks. Overall, bezel marks were significantly faster than hard button marks across environments; they had no significant difference in accuracy.

Design Recommendations	
R1	Gestural shortcuts/alternatives should be provided for soft button commands.
R2	Users should be able to assign gestures to common action sequences, e.g. “Run Phone App, Call Home” or “Run Media App, Play Classical Playlist” to make them eyes-free. The system could potentially identify such interaction patterns and automatically assign gestures to them.
R3	Mark-based gestures are faster and more accurate than free-form gestures in all the mobile environments tested, so they should be used instead of free-form path gestures unless 2D operands are required.
R4	We recommend bezel moding for design purposes as bezel marks have nearly identical performance to hard button marks; however, users preferred bezel marks.
R5	For space-critical applications, gestures could be used to save screen real estate.
R6	Because moded gestures are unlikely to be triggered by accident, they could be used to unlock the phone and execute a command, thus eliminating an extra step.

Table 4. Design Recommendations

Bezel marks were preferred (see below), and had the advantage of not requiring an additional button on the phone.

Interestingly, free-form path gestures performed significantly worse in both speed and accuracy. Users made 168% more errors with free-form path gestures than with mark-based gestures. Subsequent analysis of the gestures drawn indicates that virtually all these errors were due to user error, not a recognizer bug. We hypothesize that it is harder to perform gestures accurately one-handed with the thumb than with two hands or a stylus, which was the root cause of the errors. Users also preferred mark-based gestures to path-based.

It is notable that the specific free-form gestures used may affect performance. However, we believe they are representative of prior research systems (9 of 12 are cited), and sufficiently unique to create a robustly recognizable set.

How do various mobile environments affect the speed and accuracy of gestures? It was surprising that specific gestural conditions appeared to be unaffected by environment in speed or accuracy. Bezel marks and bezel paths, for example had no significant difference from direct to indirect, indirect to walk, or indirect to AST.

Soft buttons, in contrast, were affected by environment, and had a significant increase in completion time from direct to indirect (15.8%), after which completion time remained fairly steady. The accuracy of soft buttons was unaffected by environment. Interestingly, there was no significant difference in accuracy between soft buttons and bezel marks.

Which do users prefer? Would users like to have both soft buttons and gestures? It was surprising that for direct usage, only half the users preferred soft buttons. Furthermore, for the other 3 environments, 0 users chose soft buttons. Bezel marks were most-preferred for walking and AST, and tied with hard button marks for indirect.

Users commented almost unanimously that gestures would be useful when not devoting full attention to the phone. On gesture type, 9 commented it was difficult to perform freeform gestures; one user saying “too much thumb motion” was required, while users mentioned that rectilinear gestures helped them focus their attention on the circle centering task.

Two users felt pressing the hardware button slowed performance or felt like an “extra step.” A majority of users commented that it was easy to locate the hardware button and bezels without looking, while difficult to press soft buttons.

When asked if they would like more than one technique for different situations, 13 of 15 users chose more than one technique; just one chose soft buttons only. Based on this and the quantitative performance results, we hypothesize that gesture-shortcut alternatives to command buttons on smart phones would improve performance and reduce attentional load. Table 4 (above) outlines our design recommendations.

¹ The sphericity assumption was not met so the Huynh-Feldt correction was applied. The corrected degrees of freedom is shown.

² Beyond visual comparison against a crib sheet of gestures, the verifier marked a gesture as unknown if >20% geometry was missing, >20% new geometry was added, if a feature was rotated/skewed by >50°, or if a feature was scaled up or down by >60%. The verifier was unaware of the software recognizer’s ratings.

Other Considerations

The controlled, lab-based nature of our study limits the generality of the results. Although we simulated several types of distractions and motor activities, these may not be fully representative of ecologically valid situations. Prior experience (4 users) with touch screen devices may have affected performance; the results may not generalize to other populations. Our compensation approach may limit the generality of results. However, as remarked above, we found this was essential to prevent users from becoming “bored” and thus underperforming in speed/accuracy given the large number of trials. We do not believe it sacrificed accuracy, since users were rewarded for both speed *and* accuracy, and because our soft button accuracy of 93.9% is close to the approx. 95.0% values from [24]. Freeform path gestures may have mnemonic advantages given their greater uniqueness. Gestures are always available, whereas soft buttons must either consume screen space as a toolbar, or use an invocation procedure.

CONCLUSION

In this paper we explored the effect of mobile environments, modeled as motor activity and distraction level, on moded direct touch gestures. The results of a formal experiment indicates direct touch gestures can produce on-par performance and accuracy with soft buttons when the user is focused on the phone, and improved performance, and reduced attentional load in the presence of environmental distractions. Our results further indicate that bezel gestures did not appear to be affected by environment in speed or accuracy, and the gestures tested can be done effectively eyes free, with one hand. We found bezel-initiated gestures to be fastest, and most-preferred by users. We also found mark-based gestures were faster and more accurate to perform, and were preferred by users to free-form path gestures. We believe, therefore, that bezel-initiated, mark-based gestural shortcuts should be provided for soft button commands on mobile devices.

ACKNOWLEDGEMENTS

We wish to thank Andries van Dam for his advice and insight, Alice Liu for her illustrations, and Robert Zeleznik, Zachary Kahn, Zachary Davis, Jeff Coady, and Max Salvias for their assistance. This material is based upon work supported under a National Science Foundation Graduate Research Fellowship and by a Google Research Award.

REFERENCES

1. iSuppli. *Touch-Screen Shipments Expected to Reach 833 Million by 2013*. 2008.
2. Brewster, S. Overcoming the lack of screen space on mobile computers. *Personal and Ubiquitous Computing*, 6, 3, 188-205.
3. Sears, A., et al. When computers fade: Pervasive computing and situationally-induced impairments and disabilities. In *HCI Int'l '03*.
4. Kurtenbach, G. *The Design and Evaluation of Marking Menus*. Department of Computer Science, University of Toronto. 1993.
5. Li, K., Baudisch, P., and Hinckley, K. Blindsight: eyes-free access to mobile phones. In *Proceedings of CHI'08*, 1389-1398.
6. Charissis, V., et al. Designing a direct manipulation HUD interface for in-vehicle infotainment. In *Proc. of ICHCI'07*.
7. Kristoffersen, S., et al. “Making place” to make IT work: empirical explorations of HCI for mobile CSCW. In *SIGGROUP'99*.
8. Li, Y., Hinckley, K., Guan, Z., et al. Experimental analysis of mode switching techniques in pen-based user interfaces. In *Proc. of CHI'05*.

9. Pirhonen, A., Brewster, S., and Holguin, C. Gestural and audio metaphors as a means of control for mobile devices. In *CHI'02*.
10. Lee, S. and Zhai, S. The performance of touch screen soft buttons. In *Proc. of CHI'09*.
11. Perry, K. and Hourcade, J. Evaluating one handed thumb tapping on mobile touchscreen devices. In *Proc. of GI'08*, 57-64.
12. Lin, M., Goldman, R., Price, K., Sears, A., and Jacko, J. How do people tap when walking? An empirical investigation of nomadic data entry. *Int'l J. of Human-Computer Studies*, 65, 9, 759-769.
13. Kane, S., et al. Getting Off the Treadmill: Evaluating Walking User Interfaces for Mobile Devices in Public Spaces. In *MobileHCI'08*.
14. Roth, V. and Turner, T. Bezel swipe: conflict-free scrolling and multiple selection on mobile touch screen devices. In *CHI'09*.
15. Hinckley, K., Yatani, K., Pahud, M. et al. Manual deskterity: an exploration of simultaneous pen + touch direct input. In *CHI'10 EA*.
16. MacKay, B., Dearman, D., Inkpen, K., and Watters, C. Walk 'n scroll: a comparison of software-based navigation techniques for different levels of mobility. In *Proc. of MobileHCI'05*, 183-190.
17. Appert, C. and Zhai, S. Using strokes as command shortcuts: cognitive benefits and toolkit support. In *CHI'09*, 2289-2298.
18. Zhao, S. and Balakrishnan, R. Simple vs. compound mark hierarchical marking menus. In *Proc. of UIST'04*, 33-42.
19. Oakley, I. and Park, J. A motion-based marking menu system. In *Proceedings of CHI'07 Extended Abstracts*, 2597-2602.
20. Karlson, A., Bederson, B., et al. Understanding One Handed Use of Mobile Devices. In *Handbook of Research on User Interface Design and Evaluation for Mobile Technology*. IGI Global, 2007.
21. Bragdon, A., Zeleznik, R., et al. GestureBar: improving the approachability of gesture-based interfaces. In *CHI'09*, 2269-2278.
22. Accot, J. and Zhai, S. More than dotting the i's - foundations for crossing-based interfaces. In *Proc. of CHI'02*, 73-80.
23. Henry Dreyfus Associates. *The Measure of Man and Woman*. Whitney Library of Design, New York, 1993.
24. Perry, K. B. and Hourcade, J. P. Evaluating One Handed Thumb Tapping on Mobile Touchscreen Devices. In *GI'08*, 57-64.
25. Parhi, P., et al. Target size study for one-handed thumb use on small touchscreen devices. In *Proc. of MobileHCI'06*, 203-210.
26. Karlson, A. et al. ThumbSpace: Generalized One-Handed Input for Touchscreen-Based Mobile Devices. In *Lect. Notes, Interact'07*.
27. Kurtenbach, G. and Buxton, W. The limits of expert performance using hierarchic marking menus. In *CHI'93*, 482-487.
28. Zeleznik, R., Bragdon, A., Liu, C., and Forsberg, A. Lineogrammer: creating diagrams by drawing. In *UIST'08*, 161-170.
29. Li, Y. Protractor: A Fast and Accurate Gesture Recognizer. *CHI'10*.
30. Knoblauch, R. L., et al. Field Studies of Pedestrian Walking Speed and Start-Up Time. *Transportation Research Record* (1996), 27-38.
31. Wickens, C.D. and Hollands, J.G. *Engineering Psychology and Human Performance*. Prentice Hall, 1999.
32. Harrison, C. and Hudson, S. Providing dynamically changeable physical buttons on a visual display. In *Proc. CHI'09*, 299-308.
33. Iqbal, S., Ju, Y., and Horvitz, E. Cars, calls, and cognition: investigating driving and divided attention. In *CHI'10*, 1281-1290.
34. Holm, S. A Simple Sequentially Rejective Multiple Test Procedure. *Scandinavian Journal of Statistics*, 6 (1979), 60-65.
35. Moran, T. P., et al. Pen-based interaction techniques for organizing material on an electronic whiteboard. In *Proc. of UIST'97*, 45-54.
36. Hinckley, K., et al. InkSeine: In Situ Search for Active Note Taking. In *CHI'07*.
37. Liao, C., et al. PapierCraft: A Gesture-Based Command System for Interactive Paper. *ToCHI*, 14, 4 (2008).
38. Wobbrock, J., et al. Gestures without Libraries, Toolkits or Training: A \$1 Recognizer for User Interface Prototypes. In *UIST'07*.
39. Lepinski, G., Grossman, T., and Fitzmaurice, G. The design and evaluation of multitouch marking menus. In *CHI'10*, 2233-2242.