Tap, Swipe, or Move: Attentional Demands for Distracted Smartphone Input

Matei Negulescu [1] Jaime Ruiz [1]
[1] Cheriton School of Computer Science
University of Waterloo
Waterloo, Canada N2L 3G1
{mnegules, jgruiz, lank}@cs.uwaterloo.ca

Yang Li [2] Edward Lank [1]
[2] Google Research
1600 Amphitheatre Parkway
Mountain View, CA 94043
yangli@acm.org

ABSTRACT
Smartphones are frequently used in environments where the user is distracted by another task, for example by walking or by driving. While the typical interface for smartphones involves hardware and software buttons and surface gestures, researchers have recently posited that, for distracted environments, benefits may exist in using motion gestures to execute commands. In this paper, we examine the relative cognitive demands of motion gestures and surface taps and gestures in two specific distracted scenarios: a walking scenario, and an eyes-free seated scenario. We show, first, that there is no significant difference in reaction time for motion gestures, taps, or surface gestures on smartphones. We further show that motion gestures result in significantly less time looking at the smartphone during walking than does tapping on the screen, even with interfaces optimized for eyes-free input. Taken together, these results show that, despite somewhat lower throughput, there may be benefits to making use of motion gestures as a modality for distracted input on smartphones.

Author Keywords
Eyes-free interaction, smartphones, motion gestures.

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms
Human Factors

1. INTRODUCTION
Modern smartphone devices support two alternative input modalities. Users can tap or gesture on the touch-sensitive screen of the smartphone, or users can move the smartphone in physical space and have their actions sensed by accelerometers or gyroscopes. In our research, we are particularly interested in the costs and benefits of physical motion of the smartphone as an input modality. We call these deliberate movements of the device *motion gestures*.

Motion gestures have attractive features that recommend them as a mechanism for issuing commands on a smartphone. First, these motion gestures expand the input bandwidth of modern smartphones. For example, motion gestures can either serve as modifiers of surface gestures, or they can be mapped to specific device commands. Second, alongside the increase in bandwidth, motion gestures can represent a set of shortcuts for smartphone commands. For actions performed using the touchscreen, the phone must typically be in a specific state, e.g. a specific application must be running, or a specific toolbar must be invoked, whereas for motion gestures, the commands mapped to the gestures can be always available. Finally, motion gestures may require less visual attention than taps or gestures on the touchscreen because the physical location of the smartphone can be sensed via proprioception. As a result of the potential advantages of motion gestures for smartphone input, researchers have explored various aspects of the design of motion gesture interaction [1, 22].

One specific advantage of proprioceptive sensing is that motion gestures may be particularly beneficial as an input modality in a subset of tasks where the user is distracted while using the smartphone. There are many examples of distracted input on smartphones. For example, users frequently access email and text messages on their smartphone while walking. Therefore, users must split their attention between the task of navigating their physical environment and navigating information on the smartphone screen. As another example, users frequently invoke brief commands on their smartphones while driving. While it may be undesirable to have a user interact with their device while driving, users will continue to perform short commands. We are not the first researchers to note that it makes sense to design input techniques that demand limited visual attention from users while performing tasks like driving [5, 9, 13].

While motion gestures have many theoretical advantages as an input technique for distracted users, we are not aware of any research that compares motion gestures to on-screen input for distracted interaction. As we want to understand motion gestures for distracted input in relative to more traditional on-screen input methods, we consider surface gestures – directional *swipes* – and *taps* on pre-defined widgets. In this paper, we compare motion gestures, tap and swipe in two experimental conditions where the user has limited ability to focus on the smartphone. First, we examine user performance when the user is walking around a prescribed path and carrying a light object in their non-dominant hand. This condition replicates the situation where a user walks along a sidewalk while carrying a briefcase or purse and interacting with a smartphone. Second, we examine user performance in an eyes free setting, where the phone is not visible to the user as they perform actions. This condition replicates situations where it might be undesirable for a user to focus his or her visual attention away from their primary task, for example while driving a car. We examine reaction time, walking speed, visual focus, and throughput for the walking condition, and reaction time and throughput for the eyes-free condition.

Our experimental results show that users' response time is not significantly different for motion gestures, tap or swipe.

AVI'12, May 21-25, 2012, Capri Island, Italy.

Moreover, though participants issue significantly fewer commands with motion gestures than with tapping or flicking, and the reduction in throughput can be mainly attributed to the increased time taken for motion gestures as compared to tap or swipe. Surprisingly, we also find for the walking task that performing a motion gestures reduces the average walking speed of experimental subjects, but that subjects spend less time looking at the smartphone device when using motion gestures. Finally, we note that motion gestures are more error-prone than other techniques, and, specifically, that participant performance seems to vary over time. Recognition rates occasionally fall to very low levels, particularly during the walking task, and it is very difficult for participants to diagnose and correct the errors they are making. Our goal in presenting these results is to inform designers of motion gestures of the potential benefits and costs of motion gestures so that designers can make better decisions on where, when, and how to incorporate motion gesture into smartphones as an input technique.

The rest of the paper is organized as follows. We begin with an overview of related work in the design and evaluation of motion gestures, and in the design of limited attention interfaces. We then describe our interaction techniques and their implementation. Next, we present our experimental methodology, and describe our results. We analyze our results in terms of cognitive demand and error rates. We close with a discussion on potential limitations of motion gestures as an input modality for distracted smartphone input.

2. Related Work

Many researchers have explored interaction in distracted contexts. For example, Noy et al. found that manipulating items using touch is more cognitively demanding than traditional tactile knobs due to increased visual demand [17]. More recent work has focused on gestural interfaces on the surface of the wheel for use in distracted contexts (e.g. [2, 7, 9]). González et al. report significant performance improvements and lowered cognitive load when using the EdgeWrite eyes-free input over touch-based input [9]. Similarly, Döring et al. explored multi-touch gestures on the steering wheel and report significantly lower visual demand when compared to central console touch interaction [7].

In the mobile interaction domain, researchers have noted that, when users have to divert some of their attention to a relatively simple task like walking, their performance with the smartphone device is negatively affected. In particular Bergstrom-Lehtovirta et al. noted that there is a trade-off between walking speed and target accuracy for mobile devices [3].

To assess the relative efficacy of different interaction modalities on mobile devices during distracted tasks, Bragdon et al. examined soft buttons, hardware buttons, and surface gestures [4] under conditions of medium and high distraction. They found that marking menus (i.e. directional gestures) activated along a smartphone's bevel provided the fastest response time and the highest performance on the distractor task.

While hardware buttons, software buttons, and gestures have been the most common input modality for smartphone devices, researchers have also explored the use of accelerometers as a mechanism for issuing commands to smartphones. Recent work has considered such gestural interaction with the device for a wide variety of tasks including, text input [12, 19], issuing device commands [22], and map navigation [20].

In distracted environments, most gestures may lessen the need for visual feedback and make use of a user's proprioception to substitute for accurate input on the touch screen [18]. In an evaluation of motion marking menus, Oakley and Park note that users can access up to 19 commands accurately using three-dimensional gestures [18].

While researchers have noted the lower visual feedback of motion gestures for input, and have evaluated hardware buttons, software buttons, and gestures for distracted input, we are aware of no literature that contrasts directly the input modalities of tap and swipe to motion gestures, specifically assessing the relative cognitive costs of the different input modalities.

3. EXPERIMENT

3.1 Participants and Apparatus

We selected 12 participants aged 22-36 (mean = 25.4, S.D. = 4.8, 4 females, all right handed) from the student population in the Computer Science department at a local university.

The experiment was performed using a Nexus One smartphone running custom software on Android 2.3.3.

3.2 Experimental Design

3.2.1 Interfaces

A goal of this paper is to measure the costs of different input modalities—tap, swipe, and motion gestures—in situations where the end-user has a limited ability to visually focus on the smartphone display. The scenarios we envision include interaction during contexts such as walking or driving a car. We call this style of interaction *distracted input,* and we note that typical smartphone applications such as email clients, text message viewers, and mobile web browsers are poorly designed for contexts that require distracted input.

The guidelines for smartphone application design [8] provide little guidance for how to design interfaces for distracted input. However, the principles of interaction design for contexts where the user is distracted are relatively obvious [9]:

1. The interface should limit the need for visual attention during interaction.
2. The interface should provide streamlined commands for the most common tasks.

With these two principles in mind, we designed three alternative interfaces to support distracted input, one for tap, one for swipe, and one for motion gestures (see Figure 1 and Figure 2). Each of our interfaces supports four commands: Left, Right, Up, Down. We chose four commands for two reasons. First, the set of commands is sufficiently small that users should be able to master them within a short period of time during a training block, allowing us to measure expert performance with each input modality. Second, four commands can easily be mapped to navigation directions, Previous, Next, Up, Down, and these commands are common shortcuts for tasks such as scanning email, scanning text messages, or other monitoring tasks that are

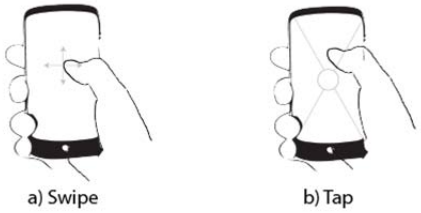

Figure 1. Distracted motion input: a) our Swipe gesture implementation and b) Tap.

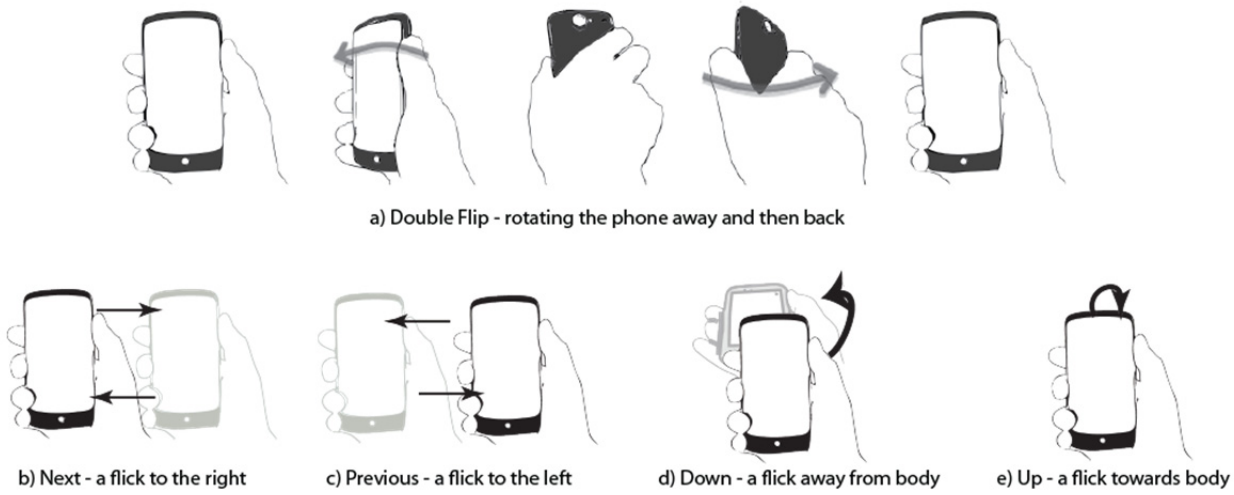

Figure 2. Our Move implementation: a) shows the Double Flip delimiter, while b)-e) show the four motion gestures.

commonly performed in distracted contexts.

When designing our experimental interface, we considered both output (screen display) and input (tap, swipe, or move). As the application was designed for situations with limited visual attention, the primary mechanism for prompting a participant for a command was a simple custom speech-to-text engine that output the command through the smartphone speakers. One of the researchers recorded each of the four commands. To ensure participants could hear the audio command clearly, we carefully tuned pitch and volume to ensure the command was easy to hear within our experimental environment. As well, the experiment was conducted in a 4m by 4m soundproof experimental room. One researcher and the participant were the only occupants of the room.

Alongside audio output, the desired command was also displayed on the screen. The visual display of the command on the screen was primarily for reinforcement. In Tap, the application display screen was a black background separated into four quadrants as shown in Figure 1b. In the non-active area in the center of the screen, a single command was displayed in 12 pt Verdana font. The Swipe (Figure 1a) and Move (Figure 2) applications consisted of a blank, black screen. As movement (either on screen or in physical space) was the sole mechanism for input, no divisions or widgets were displayed. In the center of the screen, using identical font, color, and location to Tap, the Swipe and Move interfaces displayed a single command to be activated.

For input, Tap simulates a classic widget-based approach in which the user clicks with one finger to issue one of the four commands on the touch screen. To maximize button size, the touchscreen is divided into four quadrants situated around the center of the display in manner similar to radial menus [11], as shown in Figure 1a. The user can issue the four commands necessary for our study – *up, down, left and right* – by tapping within the corresponding quadrant on the display. The area in the center of the touchscreen serves as a display for the current gesture the user needs to perform. Clicking within this small circular area does not activate any of the four commands so as to limit potential errors caused by clicking at the central intersection of the quadrants.

Swipe allows the user to perform surface gestures to issue the four commands required by the experiment (Figure 1b). Our Swipe implementation is a more permissive version of the swipe interface evaluated by Bragdon et al. [4]. In Bragdon et al., swipes were performed either along the bevel or in the center of the display. In our interface, we did not discriminate between a bevel swipe and a swipe on the touchscreen away from the bevel. Users perform a directional surface swipe gesture to activate one of up, down, left, or right, similar to the shortcuts offered by Kurtenbach et al.'s Marking Menus [14]. The Swipe recognizer considers a stroke's direction based on its starting and ending points and the largest dimension of its bounding box. In order to minimize confusion between swipe directions, the system does not recognize swipes that are less than 10px (9.9 mm) long or whose largest dimension in the bounding box is less than 3 times the smallest (i.e. it accepts a stroke if height > width x 3 or vice versa). A rejected stroke is logged as an error.

Finally, our motion gesture interface, *Move*, used four gestures from the consensus set of motion gestures described by Ruiz et al. [22]. We used a flick right for Next, a flick left for Previous, a flick up for Up, and a flick down for Down. To allow our recognizer to reliably segment these gestures from random device motion, we also made use of the Double-Flip delimiter for motion gestures proposed by Ruiz and Li [21]. To issue a command with the system, a user first performs double-flip, and then performs the appropriate motion gestures for the desired command. The five motion gestures are depicted in Figure 2. To issue a command, the end-user first performs a double-flip (Figure 2a) and then performs the appropriate motion gestures (any of Figure 2b – 2e). We implemented the Move recognizer as a Hidden Markov Model (HMM) trained with pre-segmented motion samples from five expert users.

Our Move interface has one advantage over Tap and Swipe. With Tap and Swipe, the screen's input space must be modified to support distracted input. If the screen is displaying information in the background, for example and email message, text message, or chat dialog, then the typical on-screen interactions of these applications must be disabled to support the more accessible tap and swipe gestures tailored to distracted input. In contrast, the Move interface can be designed such that the screen continues to function as an input modality without modification, and the motion gestures present a shortcut for accessing the four optimal commands. As the purpose of this paper is to contrast an optimized tap, swipe, and motion gesture interface for distracted input, we should note that we do not consider the costs associated with disabling the standard interaction. We do not claim that our

tap and swipe interfaces are real-world interfaces. Instead, they are an optimized analog to the motion gesture shortcuts, allowing us to contrast the benefits and costs of the three input modalities.

3.2.2 Experimental Tasks

Based upon previous studies that look at evaluating interaction techniques under split attention and concurrent with physical motion (e.g. [3, 4, 9], we consider two scenarios of use: i) interacting with the phone while walking and ii) in an environment with low cognitive load but where visual demand is at a premium (e.g. interacting with a phone while stuck in traffic).

Our study design was focused around the two scenarios of use:

- Walking -- Interacting while walking in our course
- Eyes-Free – Interacting with the phone held beneath a desk

As we wish to evaluate input techniques under distracted scenarios, our first scenario, walking, requires participants to perform commands while following a closed track in the soundproof room. Small arrows were placed on the floor of the room to act as a guide for participants. The course is described in Figure 3. Participants moved along walls and diagonals from position 1 to position 10 then repeated the course. Though relatively small, we found that traversing the course acted as a moderate distracter; participants had to pause at times to focus on the small floor markers telling them their next destination. During the walking task, participants held an object in one hand and performed the commands with their other hand. While not an explicit requirement, as expected all participants chose to hold the object in their non-dominant hand and to interact with the phone with their dominant hand.

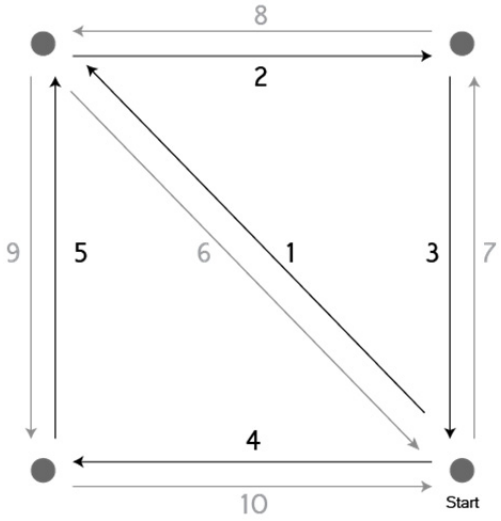

Figure 3. The route participants must traverse in the walking task. Participants walk to the corners of the room in the order given by the labels 1-10.

Our second scenario, eyes-free, was designed to mimic contexts like driving. In early pilot studies, we evaluated using distracter tasks like the Sustained Attention to Response Task (SART) test [15], an evaluation of visual attention. However, during discussion with pilot study participants, many participants noted that, when driving, they would focus on the road during cognitively demanding times, and would only partially shift their attention to a smartphone device during periods of low cognitive load (e.g. an empty street or road with no pedestrian traffic). As a result, we made a conscious design decision to eliminate all distracter tasks and to simply focus on interaction in scenarios where looking at the smartphone was not advisable. Participants performed smartphone commands with one hand beneath the table. With their other hand, participants were required to hold a light object while resting the back of their hand on the table's surface. Again, participants chose to interact with the phone using their dominant hand. The experiment was conducted in the same soundproof room, reconfigured for the seated task.

3.3 Experimental Procedure

Our experiment was a 3 X 2 (3 techniques, 2 scenarios) within-subjects design with repeated measures. The order of the techniques (Tap, Swipe, Move) was fully counterbalanced. As our goal was to compare techniques in each scenario separately, we did not counterbalance scenario ordering. No comparisons between scenarios should be drawn from our data.

Each participant began with the walking scenario. The experiment began by demonstrating all interaction techniques to participants.

Participants then completed two four-minute blocks in the walking scenario with each technique. The first block was a practice block. The goal was to familiarize participants both with the given interaction technique and with walking the track. Following a brief break, the second block tasked participants to maximize the number of correct interactions while walking at a fast but comfortable pace.

Once the walking scenario was complete, a desk was positioned in the room, and participants performed the same techniques in the seated scenario. Participants performed four minutes of gestures with each of the Move, Tap and Swipe techniques. No training block was included in the seated scenario.

As part of both scenarios, participants were asked to continually perform the available commands (e.g. one of Up, Down, Next, or Previous). The order of the commands was randomized, but all commands were performed in equal blocks of four. Each function was vocally prompted on the smartphone and any detected input, whether correct or incorrect, was recorded before the system moved to the next command in sequence.

3.4 Measures

The software on the smartphone captured the following data during the experiment:

Response Time: The amount of time (in ms) starting from the end of the vocal command prompt to the first user action. For Tap and Swipe, this time is the time at which the first touch event encountered. For Move, response time is taken from the end of the vocal command prompt to beginning of the Double-Flip delimiter.

Commands: The total number of commands issued in the four minute block.

Successful Command Rate: The fraction of commands that were recognized as the correct command by the recognizer.

During the walking scenario, the researcher also captured a set of field notes. First, the researcher manually recorded the distance traveled during the four minute block. As well, to measure visual attention during the walking task, the researcher noted the number of 5 second intervals in which participants gazed at the touch screen, sampled every 10 seconds. Lastly, the researcher made note of the number of times participants lost their way and had to reorient themselves around the track as a measure of additional cognitive load. We code the data as:

Speed: The speed of participants as measured by their distance traveled around the track within the four minute block. We transform this distance measure into meters per second by

counting the number of corners traversed, multiplying by distance between corners, and dividing by 240 seconds.

Lost: The number of times participants stopped to get their bearings and determine their next destination.

Screen Gaze: The number of five second intervals in which participants looked at the screen at least once.

4. Results

In this section, we analyze the walking scenario and sitting scenarios separately.

4.1 Walking Scenario

4.1.1 Response Time

Response time relates to the difficulty in mapping the given commands to the actions required by the interaction technique while navigating our closed course. Though we selected mappings from gestures to command from Ruiz et al.'s consensus set [22], we still expected a difference in reaction times of the Move interface when compared to the arguably more "intuitive" motion marks-like interface of Swipe and traditional widgets of Tap.

However, we found no statistical difference in response time for any of the techniques (Figure 4a). Using Move, our participants averaged 1040ms (S.D. = 628ms) to react from the audio cue to the beginning of the Double Flip gesture. In contrast, participants averaged 1037ms (S.D. = 259ms) using Swipe and had a mean response time of 954ms (S.D. =141ms) while using Tap.

An analysis of variance with technique as a within-subjects factor did not find a significant effect for the differences seen in response time ($F_{2,22}$ = 0.352, ns). The large standard deviation while using Move seemed primarily an effect of the variability in *Success Rates* with the gestures. Participants P3, P5 and P12 had significant trouble performing motion gestures (i.e. had very low *Success Rates)*, and had correspondingly high *Response Times* (mean response time for P3, P5 and P12 was 1950ms).

4.1.2 Command Throughput

Throughout the four-minute sessions participants were asked to perform as many accurate command activations as possible. As a result, the command throughput can be estimated by considering how many commands participants attempted to perform, whether successful or otherwise, in the four minute session.

Figure 4b-c shows a summary of our walking condition's Total Commands attempted and associated Success Rate. Participants attempted 63.2 commands (S.D. = 11.0) using our motion gestures in Move. In contrast, participants attempted 146.5 total commands with Swipe (S.D. = 18.9) and 164.3 commands using the Tap interface (S.D = 16.2).

First, we note that an analysis of variance with technique as a within-subjects factor found a significant effect on Total Commands issued in the walking scenario ($F_{2,22}$ = 374.45, $p < 0.001$). Post-hoc analysis using Bonferroni correction showed that the difference in total commands attempted was significant: participants issued significantly fewer commands while using Move than with Tap or Swipe ($p < 0.001$). Moreover, post hoc analysis also demonstrated that participants issued fewer total commands with Swipe than with Tap ($p < 0.05$).

Secondly, an analysis of variance with technique as a within-subjects factor similarly found that technique had a significant effect on Successful Command Rate ($F_{2,22}$ = 40.017, $p < 0.001$). Post-hoc analysis using Bonferroni correction showed significant differences between success rate of Move and Swipe ($p < 0.001$), Move and Tap ($p < 0.001$) and Swipe and Tap ($p < 0.05$). Participants performed significantly worse in terms of success rate with Move (M = 0.73, S.D. = 0.11) than with Swipe (M = 0.89, S.D. = 0.07) or Tap (M = 0.97, S.D. = 0.02).

4.1.3 Physical Characteristics

Our last metrics – Screen Gaze, Speed, and Lost – give us insight into the effects each technique has on user behaviors during the walking scenario.

We first report mean Screen Gaze – the number of times participants looked at the device`s display, sampled for five seconds every ten seconds – in Figure 4d. An analysis of variance with technique as a within-subjects factor found a significant effect on Screen Gaze ($F_{2,22}$ = 4.34, $p < 0.05$). Post-hoc analysis using Bonferroni correction showed significant differences between the numbers of times participants gaze at the display with Move when compared to Tap ($p < 0.01$). Specifically, participants

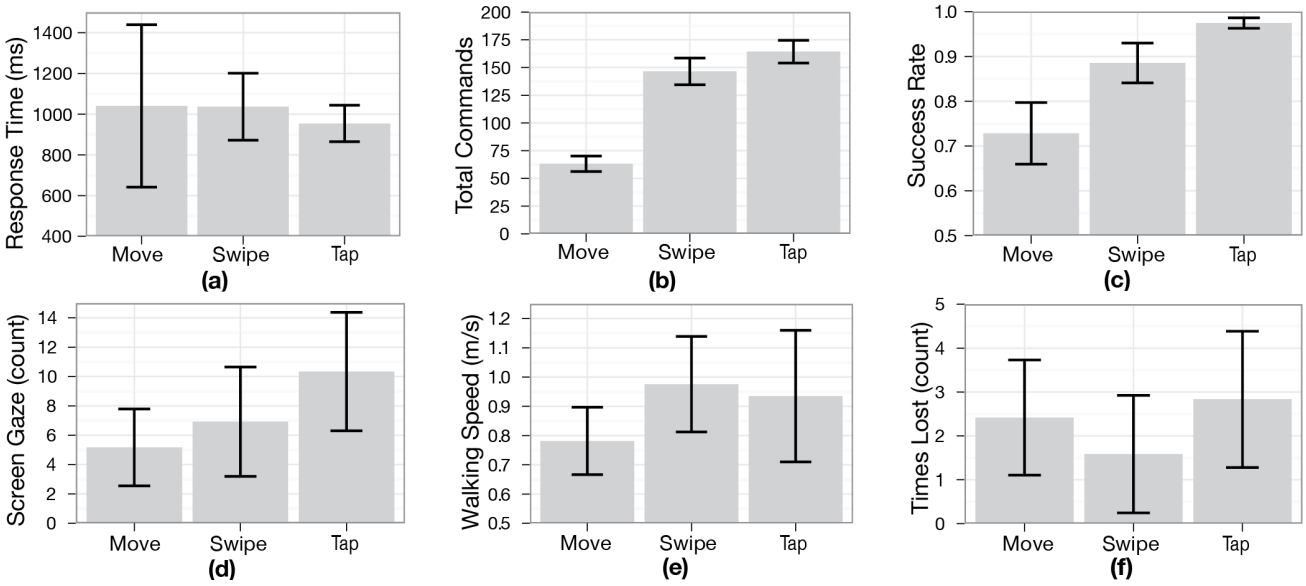

Figure 4. Summary of the Walking Scenario: (a) mean response time, (b) mean total commands attempted, (c) mean success rate, (d) mean number of screen gazes, (e) mean walking speed in m/s, and (f) mean number of times lost.

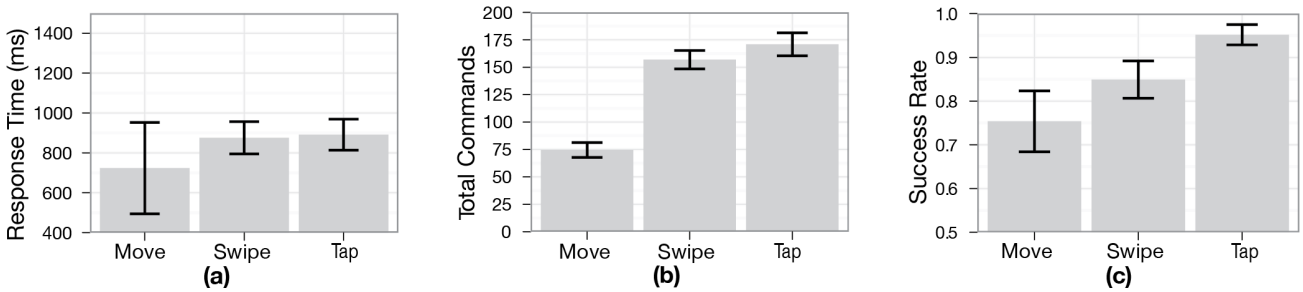

Figure 5. Eyes-free Scenario: (a) mean response time, (b) mean total commands attempted, and (c) mean success rate.

watched the display significantly less with Move (M = 5.2, S.D. = 4.13) compared to Tap (M=10.3, S.D. = 6.4). No other significant differences were found.

Secondly, Figure 4e shows an aggregate of walking speeds using the three interaction techniques. Analysis of variance shows a significant effect of technique on Speed ($F_{2,22} = 15.85$, $p < 0.001$). Post-hoc analysis using Bonferroni correction showed participants walked significantly ***slower*** while using Move than while using Tap ($p < 0.05$). Participants walked at an average speed of 0.8m/s. (S.D. = 0.18m/s) with Move compared to the brisker pace of 1m/s and 0.9m/s while using Swipe and Tap respectively.

Finally, Figure 4f shows a summary of mean times Lost (that is, the number of times participants had to stop and reorient themselves). Participants consistently got lost an average of two times regardless of interaction technique used ($F_{2,22} = 1.45$, $p > 0.2$). As expected, the majority of instances where participants stopped to reorient themselves happened during their first session in the Walking task, regardless of technique used.

4.2 Eyes-free Scenario

For the eyes-free scenario, we performed an analysis of variance with technique as a within-subjects factor to find significant effects of technique on response time, total commands, and success rate. The results, shown in Figure 5, mirror those in our walking scenario and can be summarized as follows:

1. There was no significant effect of technique on response time ($F_{2,22} = 2.580$, $p > 0.05$).
2. Technique significantly affected Total Commands issued ($F_{2,22} = 254.84$, $p < 0.001$). Post-hoc analysis using Bonferroni correction showed that participants issued significantly fewer commands with Move than with either Swipe ($p < 0.001$) or Tap ($p < 0.001$) .Additionally, participants issued fewer commands with Swipe than with Tap ($p < 0.05$).
3. Again, as expected, technique significantly affected Success Rate ($F_{2,22} = 23.616$, $p < 0.001$). Post-hoc analysis using Bonferroni correction found significant differences between the success rate of participants using Move and Tap ($p < 0.001$), Swipe and Tap ($p < 0.05$) and between Move and Swipe ($p < 0.05$). Success rate was significantly lower for Move (M = 0.75, S.D. = 0.11) when compared to both Swipe (M = 0.85, S.D. = 0.07) and Tap (M = 0.95, S.D. = 0.04).

In all, participants did not change their behavior while interacting in a stationary eyes free scenario from that of walking. Participants performed significantly fewer commands with Move (M = 74.5, S.D. = 10.6) than with either Swipe (M= 156.8, S.D. = 13.2) or Tap (M= 170.8, S.D. = 16.4), though their response rates were not significantly different. Additionally, the significantly lower success rate of 75% for motion gestures in Move mirrors the success rate for the walking task.

5. Discussion

In this section, we first address the cognitive cost of motion gestures. We then examine the issues of command throughput and recognition. In our discussion, we focus specifically on the design implications of the findings on cognitive cost and throughput.

5.1 Cognitive Cost

Our evaluation of motion gestures, tap, and swipe as input modalities falls into the broad category of psychometric studies known as *Recognition Reaction Time* experiments [23]. In these experiments, an experimental subject responds to a stimulus by planning and initiating a sequence of actions. The time between the end of the stimulus and the initiation of the response is the reaction time.

Many personal factors can affect deviations in reaction time—illness, skill, gender, age, handedness, fatigue, distraction. However, for a given controlled environment, reaction time is the accepted measure of the relative cognitive complexity or cognitive cost of different tasks [23]. All other factors being equal, a longer reaction time implies a task that is more complex. One specific example of the relationship between reaction time and cognitive cost that has been leveraged by HCI researchers is Hick's Law [6, 10] as a model of menu-selection complexity.

To control for personal factors, we counterbalanced order of techniques (Tap, Swipe, Move) and used a within-subjects experimental design. As a result, any differences in reaction time observed between the techniques are the result of a longer planning phase before onset of action. Techniques with a longer planning phase are considered to have higher cognitive cost.

In our results section, we note that there was no significant difference in reaction times while using Move, Swipe or Tap. Statistically, users were equally able to successfully build mental models that link physical movements with the device with the given shortcut commands as they were to map tapping or swiping actions to commands. There are two possible interpretations of this datum. The first is that there is no additional cognitive complexity associated with motion gestures. The second is that, although there is a difference, the difference is too small to be measured given the inherent noise associated with different participants' reaction times.

Regardless of which interpretation is true, the design implications of this finding are that motion gestures are a potentially beneficial input modality for distracted scenarios. The difference in cognitive cost between motion gestures, tap, and swipe is sufficiently small. Furthermore, because of other benefits associated with motion gestures – always-available, proprioceptive sensing – the lack of observable increase in cognitive cost is a positive result.

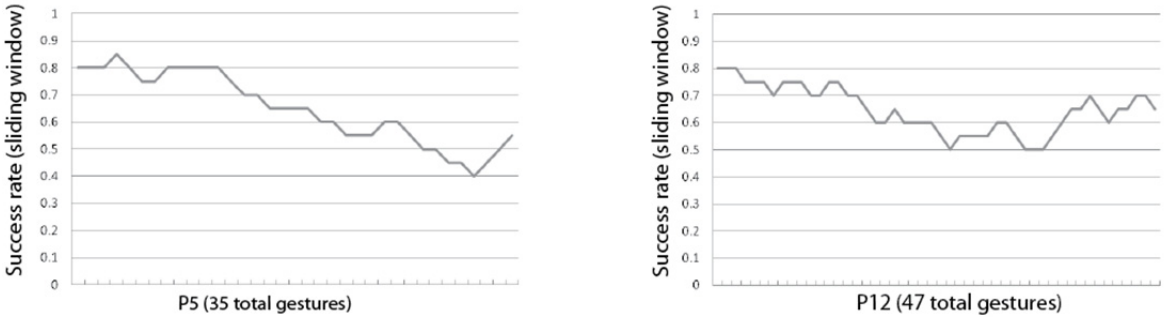

Figure 6. Success rate of Move over a five-command sliding window of P5 (left) and P12 (right)

While reaction times were the same, we found that participants walked significantly slower with motion gestures than with Tap in the Walking scenario. This result surprised us, particularly because during the Move technique, participants spent significantly less time looking at the interface than during Tap.

This trend became apparent very early in our experiment. As a result, we observed participants and attempted to catalogue why participants speed varied. We noted that participants speed was particularly affected by the recognition rate of the Move interface. Throughout the experimental block, recognition rates would vary naturally. As consecutive failures in recognition accumulate, we found that participants devoted higher attention to attempting to dislodge themselves from the performance trough. For instance, consider Figure 6 which outlines the Move interface`s success rate over a five-command sliding window of participants P5 and P12. Recognition performance begins at a relatively high rate of .8 then slowly drifts down to levels at or below .5 before improving again over the four-minute block. As these participants approach these minima of performance, our field notes suggest that they slowed down and took greater care in their motion gestures. Additionally, participants gazed at the screen more frequently during these periods of poor performance.

From a design perspective, this does raise questions about recognizer reliability. In our experiments, we considered simulating reliably high recognition rates; this requirement introduces confounds into the data. Motion gestures are constrained by the reliability of recognizers we can design. As a result, any realistic evaluation of motion gestures must include the effect that natural variations in motion gesture performance will have on end-user behavior.

While building a cognitive model of motion gestures does not appear to be difficult for our users, the lower speed seems to indicate that, when recognition errors occur, our participants slow down and look at the device more often so that they can perform motion gestures more carefully. Moreover, this trend seems to decrease as their performance improves.

5.2 Interaction Throughput and Recognition

While motion gestures can be used as an always-available shortcut to commands and can be invoked via proprioception without requiring a user to gaze at the display screen, motion gestures do take longer to perform than either a tap or a swipe.

To issue any one of the four commands with motion gestures, participants would perform a double-flip delimiter followed by the appropriate motion gesture. Both the delimiter and the specific motion gesture took, on average, 650ms. As a result, a motion gesture consumed approximately 1.5 seconds, including a brief pause between delimiter and specific command gesture. Because taps and swipes consume only a fraction of a second, we would expect that Move throughput would be lower than Tap or Swipe.

Beyond the longer time required to perform motion gestures, we also expect that motion gestures will suffer from recognition inaccuracies which will negatively impact throughput. Tap requires only location mapping, and Swipe can be recognized by a simple decision tree. In contrast, a motion gesture, measured imperfectly by accelerometer data, requires a carefully trained Hidden Markov Model or other trained template recognizer.

In our observations, we expected that recognizer reliability would be poorer for motion gestures. However, the variability in recognition rate and its effect on walking speed was something we did not anticipate. In analyzing our field notes, one theme that emerged from recognizer error was that repeated errors compounded problems associated with motion gesture input. Participants who experienced repeated failures would try to diagnose why those recognition failures were occurring. In the case of taps and swipes, diagnosing why an action failed was relatively straight-forward. However, in the case of motion gestures, participants had no way to characterize why gestures failed. As a result, they would try to vary the intensity, the timing, the direction, the device angle, etc. In essence, users tried to explore the space of recognizer inputs to determine whether some other set of parameters of movement would enhance accuracy.

There are several design implications that can be drawn from our observations of motion gesture interaction. Overall, these can be separated into implications for recognizer feedback, recognizer design, and motion gesture input. First, from the perspective of recognizer feedback, it would be beneficial to design techniques to communicate to users the parameters of input motion being observed by the device's accelerometers and a comparison between those parameters and the parameters associated with a specific motion gesture. Then, if users fail to activate a motion gesture, they can potentially stop, observe the desired parameters of their command, contrast with what the phone is observing, and more accurately diagnose recognition problems.

Second, from the perspective of recognizer design, it may be possible to build recognizers that prevent repeated errors by adapting in various ways to the end-user. Our recognizer was built upon a set of models that were learned from expert users. However, additional models that are more permissive may increase the reliability of recognition for end-users. For example, Negulescu et al. have explored modifying thresholds for successful motion gesture activation based on repeated actions of users [16]. In this work, if two similar inputs are observed, a more permissive model with lower thresholds is used to see if the two failed attempts could reasonably map to a specific gesture.

Finally, for any input modality, there are trade-offs. While throughput is significantly lower for motion gestures, the cost of

the motion gesture must be balanced against the benefits of always-available, eyes-free command activation. Users of smartphone systems have been willing to trade off physical keyboards on smartphones for smaller, more aesthetically pleasing profiles that use on-screen keyboards. The use of on-screen keyboards, however, slows text entry. Users have also been willing to accept four- or five-inch screens for tasks like web-browsing and email in place of larger displays. This promotes portability, but also slows email browsing and reading. From interviews and field notes with our participants, it seems that users may also be willing to accept the lower throughput and recognition errors of motion gestures if, alongside these costs, the benefits of constant availability and eyes-free input are preserved. Motion gestures are not a panacea for every potential input problem faced by end-users, but, in distracted contexts, they can serve a valuable purpose as an alternative modality.

6. Future Work

Obviously, a better recognition algorithm would disproportionately benefit motion gesture interaction. However, recognition algorithms are complex for motion gestures particularly because the actual movement of the device is a hidden model, observed imperfectly through accelerometers, and then recognized from noisy input data caused by walking or holding the smartphone. We continue to explore ways to increase the reliability of recognition, including experiments with adapting thresholds and experiments with techniques like camera-based optical flow as another data point for our recognition algorithms.

Beyond enhancements in recognition, for any novel input technique there is a question of long-term acceptance and use. Some current smartphone apps support a restricted set of motion gestures. For example, the Google App for iPhone makes use of proximity and movement to turn on the microphone when a user brings the smartphone to their ear. However, end-users seem unaware of the existence of these motion gestures, and it is not clear whether or not they are used. If motion gestures were mapped onto a set of shortcut commands, and if they were consistently available, end-user behavior might change. We hope to leverage existing contexts with smartphone system software providers to experiment with always-available input mechanisms.

7. Conclusion

In this paper, we analyze the relative cognitive cost of motion gestures, tap and surface gestures as input for smartphone devices under conditions of light distraction. We show that, for both walking and eyes-free input, the cognitive cost of motion gestures (measured as a function of reaction time) is statistically indistinguishable from the cognitive costs of taps and gestures. As a result, motion gestures represent a viable input alternative for situations where eyes-free input may be required.

8. ACKNOWLEDGMENTS

Funding provided by the Natural Science and Engineering Research Council of Canada (NSERC) and the Networks of Centres of Excellence for Graphics, Animation and New Media (NCE-GRAND).

9. REFERENCES

[1] Ashbrook, D. and Starner, T. 2010. MAGIC: A Motion Gesture Design Tool. *CHI '10: Proc. of Human factors in computing systems* (Atlanta, Georgia, Apr. 2010), 2159—2168.

[2] Ba h, K.M. et al. 2008. You can touch, but you can't look: interacting with in-vehicle systems. *Proc. of Human factors in computing systems* (New York, NY, USA, 2008), 1139–1148.

[3] Bergstrom-Lehtovirta, J. et al. 2011. The effects of walking speed on target acquisition on a touchscreen interface. *Proc. of Human Computer Interaction with Mobile Devices and Services* (New York, NY, USA, 2011), 143–146.

[4] Bragdon, A. et al. 2011. Experimental analysis of touch-screen gesture designs in mobile environments. *Proc. of Human factors in computing systems* (New York, NY, USA, 2011), 403–412.

[5] Christiansen, L.H. et al. 2011. Don't look at me, i'm talking to you: investigating input and output modalities for in-vehicle systems. *Proc. of conference on Human-computer interaction - Volume Part II* (Berlin, Heidelberg, 2011), 675–691.

[6] Cockburn, A. et al. 2007. A predictive model of menu performance. *Proc. of Human factors in computing systems* (New York, NY, USA, 2007), 627–636.

[7] Döring, T. et al. 2011. Gestural interaction on the steering wheel: reducing the visual demand. *Proc. of Human factors in computing systems* (New York, NY, USA, 2011), 483–492.

[8] G, E. and Nilsson 2009. Design patterns for user interface for mobile applications. *Advances in Engineering Software*. 40, 12 (2009), 1318 - 1328.

[9] González, I.E. et al. 2007. Eyes on the road, hands on the wheel: thumb-based interaction techniques for input on steering wheels. *Proc. of Graphics Interface 2007*, 95–102.

[10] Hick, W.E. 1952. On the rate of gain of information. *The Quarterly Journal of Experimental Psychology*. 4, (1952), 11–26.

[11] Hopkins, D. 1991. The design and implementation of pie menus. *Dr. Dobb's J.* 16, 12 (Dec. 1991), 16–26.

[12] Jones, E. et al. 2010. GesText: Accelerometer-based Gestural Text-Entry Systems. *CHI '10: Proc. of conference on Human factors in computing systems* (Apr. 2010).

[13] Kern, D. et al. 2009. Writing to your car: handwritten text input while driving. *Proc. of conference extended abstracts on Human factors in computing systems*, 4705–4710.

[14] Kurtenbach, G. and Buxton, W. 1994. User learning and performance with marking menus. *Proc. of the conference on Human factors in computing systems: celebrating interdependence* (New York, NY, USA, 1994), 258–264.

[15] Manly, T. et al. 1999. The absent mind: further investigations of sustained attention to response. *Neuropsychologia*. 37, 6 (Jun. 1999), 661-670.

[16] Negulescu, M. et al. 2011. A Recognition Safety Net: Bi-Level Threshold Recognition for Mobile Motion Gestures. *Proc. of MobileHCI MobileGestures extended abstracts (2011)*.

[17] Noy, Y.I. et al. 2004. Task interruptability and duration as measures of visual distraction. *Applied Ergonomics*. 35, 3 (2004), 207 - 213.

[18] Oakley, I. and Park, J. 2009. Motion marking menus: An eyes-free approach to motion input for handheld devices. *International Journal of Human-Computer Studies*. 67, 6 (2009), 515 – 532.

[19] Partridge, K. et al. 2002. TiltType: accelerometer-supported text entry for very small devices. *UIST '02: Proc. of symposium on User interface software and technology*, 201–204.

[20] Rekimoto, J. 1996. Tilting operations for small screen interfaces. *UIST '96: Proc. of symposium on User interface software and technology*, 167–168.

[21] Ruiz, J. and Li, Y. 2011. DoubleFlip: A Motion Gesture for Mobile Interaction. *Proc. of conference on Human factors in computing systems* (Vancouver, British Columbia, 2011).

[22] Ruiz, J. et al. 2011. User-defined motion gestures for mobile interaction. *Proc. of conference on Human factors in computing systems* (New York, NY, USA, 2011), 197–206.

[23] Welford, A. T. 1980. *Reaction Times*. Academic Press.