Cluster Touch

Improving Touch Accuracy on Smartphones for People with Motor and Situational Impairments

Martez E. Mott and Jacob O. Wobbrock
Information School | DUB Group
University of Washington
Seattle, WA USA 98195-2840
{memott, wobbrock}@uw.edu

ABSTRACT

We present Cluster Touch, a combined user-independent and user-specific touch offset model that improves the accuracy of touch input on smartphones for people with motor impairments, and for people experiencing situational impairments while walking. Cluster Touch combines touch examples from multiple users to create a shared user-independent touch model, which is then updated with touch examples provided by an individual user to make it user-specific. Owing to this combination, Cluster Touch allows people to quickly improve the accuracy of their smartphones by providing only 20 touch examples. In a user study with 12 people with motor impairments and 12 people without motor impairments, but who were walking, Cluster Touch improved touch accuracy by 14.65% for the former group and 6.81% for the latter group over the native touch sensor. Furthermore, in an offline analysis of existing mobile interfaces, Cluster Touch improved touch accuracy by 8.21% and 4.84% over the native touch sensor for the two user groups, respectively.

CCS CONCEPTS

• Human-centered computing → Accessibility technologies.

KEYWORDS

Ability-based design; accessibility; motor impairments; touch input; touch modeling; smartphones; situational impairments.

ACM Reference format:

Martez E. Mott and Jacob O. Wobbrock. 2019. Cluster Touch: Improving Touch Accuracy on Smartphones for People with Motor and Situational Impairments. In *2019 CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4–9, 2019, Glasgow, Scotland, UK.* ACM, New York, NY, USA. Paper 27, 14 pages. https://doi.org/10.1145/3290605.3300257

ACM ISBN 978-1-4503-5970-2/19/05...$15.00.
DOI: https://doi.org/10.1145/3290605.3300257

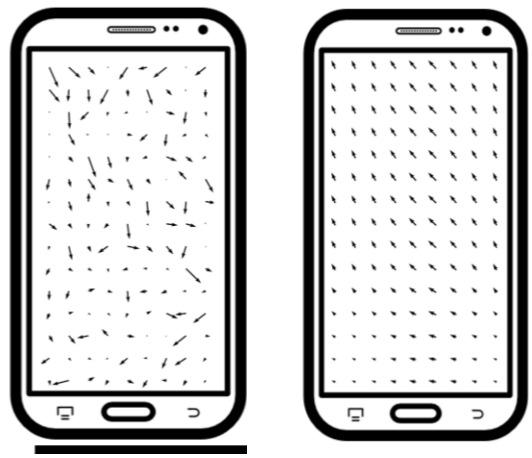

Figure 1: Cluster touch takes touch examples from individuals (left) and combines them with a user-independent model to create a user-specific touch model (right). Arrows convey corrective touch offsets by region.

1 INTRODUCTION

Touch input is one of the most ubiquitous forms of interacting with computing systems. As an input method, touch is direct, fast, and intuitive, which makes it popular among users. The popularity of touch input can be attributed to the proliferation of touch-enabled devices such as smartphones, tablets, wearables, public kiosks, and interactive wall displays.

Smartphones are one of the most pervasive touch-enabled devices in the world today. Smartphones offer many benefits to users, like the ability to communicate with friends and family, complete work on-the-go, and access the internet while away from the home or office. For people with motor-impairing conditions like cerebral palsy or muscular dystrophy, smartphones can also provide a sense of freedom and empowerment [31,42]. However, many users with motor-impairing conditions encounter accessibility challenges when interacting with their smartphone touch screen [22,23,42,51]. Inaccurate touch input on smartphones can make it difficult to perform everyday tasks such as sending text messages, composing emails, taking photos, or playing games [39,42,51]. Users undergoing the effects of situational impairments [49,50]—induced by activities such as walking—also experience difficulties with touch accuracy [3,20,32,33]. Actions like walking induce vibrations that can cause users to be less accurate when touching the screen, resulting in more input errors [3,20].

Prior research in lab [10,14,16,22,23,30,40,51] and field [1,31,38,42] settings have highlighted the numerous accessibility challenges experienced by people with motor impairments when interacting with touch-enabled devices. However, our understanding of the magnitude and direction of touch offsets made by users with motor impairments, and users undergoing the effects of situational impairments, is limited. Understanding offset errors will allow us to implement touch models that can accommodate and correct users' touches, resulting in more accurate touch input.

To improve the accuracy of touch input on smartphones for people with motor impairments, and people without motor impairments undergoing the effects of situational impairments, specifically walking, we present: (1) an exploration of the touch error offsets committed by four users with motor impairments, and eight users without impairments while standing and walking, and (2) a combined user-independent and user-specific touch model called *Cluster Touch*. In our touch-offset exploration, we found that the direction of touch offsets made by users with and without motor impairments were similar, but the magnitude of the errors was more pronounced for users with motor impairments, and for users without motor impairments while walking, than for stationary people.

Cluster Touch consists of a user-independent model that combines touch examples from multiple users to learn smartphone touch behaviors across all regions of the screen. Cluster Touch combines the user-independent model with examples from a given user to make the model user-specific, allowing it to more accurately correct touch offsets generated by that user (see Figure 1). The advantage of Cluster Touch is that it allows users to calibrate their smartphone touch screens quickly by only providing ~20 touch examples. This advantage is significant for people with motor impairments, who often fatigue quickly and for whom extensive calibration can be a deterrent.

To evaluate Cluster Touch, we conducted three separate evaluations. First, we conducted an interactive target-selection study with 12 participants with motor impairments and 12 participants without impairments, but who were walking. Cluster Touch improved touch accuracy by 14.65% and 6.81% over the native touch sensor for the two user groups, respectively.

Second, in an offline analysis of predicted touch accuracy, Cluster Touch was 20.20% and 5.61% more accurate than the native touch sensor in predicting intended touch locations for users with and without motor impairments, respectively. Also, we compared these results to two prior statistical machine learning models of touch designed to improve touch accuracy [9,55]. We found that Cluster Touch was 11.36% more accurate than the Gaussian Process model [55] and 10.79% more accurate than the Linear Offset model [9] for participants with motor impairments. For participants without motor impairments who were walking, Cluster Touch was 4.64% more accurate than both the Gaussian Process and Linear Offset models.

Third, we evaluated Cluster Touch in an offline analysis of predicted touch accuracy on existing smartphone user interfaces. We found that Cluster Touch improved touch accuracy by 8.21% and 4.84% over the native touch sensor for users with and without motor impairments, respectively.

The contributions of this work are: (1) an examination of the touch offsets of people with motor impairments, and of people without motor impairments who are standing and walking; (2) Cluster Touch, a combined user-independent and user-specific touch model that corrects touch offsets on smartphones; (3) empirical results from three separate evaluations of Cluster Touch, two offline and one interactive, all showing improvements over the native touch sensor for people with motor impairments and people who are walking; and (4) empirical results comparing Cluster Touch to two prior statistical machine learning models of touch [9,55]. This work advances our understanding of the touch behaviors of people with motor impairments, and of people undergoing the effects of situational impairments, and provides an ability-based design [57,58] approach for improving the accuracy and accessibility of smartphone touch screens.

2 RELATED WORK

This work adds to, and builds upon, prior research on touch screen accessibility for people with motor impairments, probabilistic and statistical models of touch input, theories of touch input, the effects of situational impairments, and ability-based design. These topics are briefly covered below.

2.1 Touch Screen Accessibility for People with Motor Impairments

Understanding and improving the accessibility of touch screens has been a mission for many researchers over the years, and researchers have investigated different aspects of touch screen use. Some research has focused on users' lived experience, seeking to understand how people with motor impairments use and interact with touch screens in their daily lives [1,31,38,39,42]. In these investigations, researchers found that mobile touch-enabled devices can provide a sense of freedom and empowerment to users with motor impairments [31,42].

In addition to naturalistic explorations of touch screen accessibility, researchers have also investigated touch screen use in lab settings [10,14,17,22,23,30,40,51,59]. Trewin *et al.* [51] found that users with motor impairments experienced difficulties when performing sliding and tapping gestures, which often resulted in more errors and the accidental activation of other smartphone features, such as zooming. Guerreiro *et al.* [22,23] investigated how well users with motor impairments could perform different interaction techniques. The authors found that users' performance varied depending where on the screen the user was targeting. Our work builds upon this prior research by understanding the touch error offsets created by users with motor impairments while touching the screen, and corrects those errors with Cluster Touch.

Other researchers have created new algorithms and input techniques to improve the accessibility of touch input for people with motor impairments. Biswas and Langdon [6] devised a set of models based on an analysis of users' hand strength to improve pointing accuracy on different input devices. Montague *et al.* [38] created a novel tap-gesture recognizer that uses probabilistic instead of heuristic criteria to identify taps made by users with motor impairments. Montague *et al.* [37] also created a *Shared User Modeling Framework* that adjusts the location and size of user interface elements to best accommodate the touch behavior of the user. Wacharamanotham *et al.* [53] introduced *Swabbing*, an interaction technique designed for older adults that allows users to drag their finger across the screen and lift to select their intended target. Mott *et al.* [40] created *Smart Touch*, a template-matching algorithm that allows users to touch using whichever part of their hand they choose. A goal of Cluster Touch is to allow users to calibrate their smartphone touch screens quickly, improving touch input accuracy without changing the underlying interface.

2.2 Probabilistic and Statistical Models of Touch

Touch is an inherently uncertain form of input. Because a touch represents an entire area—unlike a mouse cursor, whose hotspot occupies only a single pixel—the true location a user intends to touch is left to the system to interpret. Researchers have devised methods and techniques to leverage the uncertain nature of touch input to improve touch accuracy. Schwarz *et al.* [48] created an architecture, a framework [46], and new methods [47] for handling uncertain inputs such as touch. *Bayesian Touch* [5] by Bi and Zhai is a target acquisition technique that abandons the commonly used bounding box selection heuristic in favor of a statistical selection method based on Bayes' rule and the bivariate Gaussian distribution principle of finger touch [4]. *ProbUI* [8] by Buschek and Alt replaces static target bounding regions with probabilistic gestures, allowing users to perform a wide array of actions on otherwise static targets. Probabilistic methods have shown great promise in improving touch accuracy, but they require systems to be *target-aware* [2], meaning the system must be aware of the size, location, and state off all on-screen targets—a difficult feat to accomplish both practically and theoretically [13].

Other researchers have treated touch input as a machine learning problem and have used different algorithms to improve touch accuracy. Weir *et al.* [54,55] used *Gaussian Process* regression to build user-specific models of touch. Buschek *et al.* [9] created a machine learning approach for training user-specific touch models that work across different devices. *TouchML* [7] by Buschek and Alt is a machine learning toolkit that provides access to different touch-offset models. Touch-offset models based on machine learning have been shown to be effective for improving touch accuracy, but they often require hundreds of touch examples to reach their best performance [55]. Providing hundreds of touch examples can be problematic for people with motor impairments, as they typically take longer to input touches and experience fatigue faster than users without motor impairments. Cluster Touch creates a user-specific model with only 20 touch examples, which allows users to calibrate their smartphone touch screen quickly and without exerting much effort.

2.3 Theories of Touch Input

Holz and Baudisch [28,29] conducted two studies to understand users' perceptions of touch. In their first study, they introduced the *general perceived input point model* [28], which claimed that the primary source of touch input error is not caused by the *fat finger problem* [52], but is caused by users' perception of touch input, which creates systematic errors when using touch. They found that applying a reverse touch offset could correct erroneous touch points. In their second study [29], the authors explored different touch models based on visible finger properties. Again, the authors found that applying a corrective touch offset could improve accuracy. Henze *et al.* [25] conducted a large scale analysis of touch input gathered from a mobile game. The authors found that the direction of touch error offsets varied depending on the region of the screen. Together, these studies show that users introduce systematic touch errors, and that a reverse offset can correct these errors. Cluster Touch builds on these prior efforts by examining and correcting touch offsets created by people with motor impairments, and people undergoing the effects of situational impairments.

2.4 Situational Impairments and Touch

Situationally-induced impairments and disabilities, or "situational impairments" for short, are caused by environmental factors that temporarily affect users' abilities to interact with computing devices [49,50]. Everyday activities such as walking or carrying a bag of groceries can have adverse effects on touch accuracy [3,11,44]. Lin *et al.* [33] found that tapping accuracy was significantly reduced while walking. Kane *et al.* [32] proposed *walking user interfaces* to enlarge targets while users walk to maintain touch accuracy. *WalkType* [20] by Goel *et al.* is a technique to improve text entry on mobile devices by leveraging accelerometer data. Musić *et al.* [41] showed that users' gait phase can be used as a signal for an offset model to improve touch accuracy. In our work, we demonstrate that technology developed to aid people with motor impairments also benefits users in motor-impairing situations.

2.5 Ability-Based Design

Ability-based design [57,58] is a set of concepts, principles, and examples that implore designers to focus on the abilities that users possess, rather than on what abilities users lack. Ability-based design encourages designers to place the burden of adapting to users' abilities on systems, rather than force users to adapt themselves to meet the ability-demands of the technologies they use. The SUPPLE system [18,19] embodies this principle well, as SUPPLE automatically generates customized user interfaces based on a user's mouse-pointing performance. Cluster Touch embraces ability-based design by allowing users to calibrate their smartphone touch screens to better accommodate their touch behaviors, rather than having to adapt themselves to the demands of existing touch screens.

3 EXPLORATION OF TOUCH OFFSETS

Touch input is imprecise because a finger occupies an area instead of a single pixel [52], and because users have their own mental models of how finger contact corresponds to their desired touch-point on the screen [28,29]. When a user attempts to acquire a target, the distance between the reported touch location and the location of the intended target is a two-dimensional touch-offset vector (Figure 2).

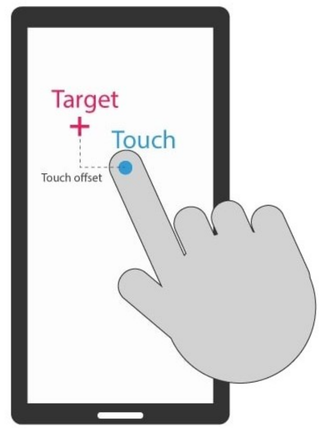

Figure 2: The difference between a user's touch location and the location of the intended target is a two-dimensional touch-offset vector.

Prior research has investigated touch error offsets [9,25,28,29,55]. Our work adds to prior work by examining the touch error offsets created by people with motor impairments, and people without motor impairments experiencing situational impairments. Without the knowledge of how touch offsets affect these users' touch accuracies, it is difficult to devise solutions to correct for these errors. To better understand these touch offsets, we conducted a preliminary study where we collected touch data from users with and without motor impairments repeatedly touching a randomly-placed crosshairs displayed on a Google Nexus 6 smartphone.

3.1 Participants

We recruited 4 people with motor impairments (3 female, 1 male, average age 31.5, *SD*=6.8), and 8 people without impairments (5 female, 3 male, average age 27.4, *SD*=5.1) to participate in our exploratory study. Participants with motor impairments were paid $30 USD for their participation, and participants without motor impairments were paid $15. Participants were recruited through word of mouth and through listservs of organizations that support people with various motor impairments. All participants were right-handed. Two participants with motor impairments had cerebral palsy, one had muscular dystrophy, and one had multiple sclerosis.

3.2 Apparatus

Touch data was collected using an experiment testbed developed for Android 8.1 using C# and the Xamarin framework. The testbed captured and logged all touch events. For each touch event, the reported *x*- and *y*-coordinates of the touch were captured along with a timestamp. All sessions were conducted on a Google Nexus 6 smartphone (15.14 cm diagonal with a screen resolution of 1440 × 2560 pixels) running Android 8.1.

3.3 Procedure

Each participant completed a single lab session that lasted between 30 minutes to an hour. Participants with motor impairments completed 540 target selection trials in each session while seated. Non-motor-impaired participants completed 540 selections trials while standing, and another 540 selection trials while walking on a treadmill. The treadmill's speed was set to allow for a comfortable walking pace for each participant. Non-motor-impaired participants were asked to hold the phone in their non-dominant hand, and to touch the screen with the index finger of their dominant hand. Participants with motor impairments were asked to hold the phone in whichever way was most comfortable for them, and to touch the screen with a finger on their dominant hand (Figure 3).

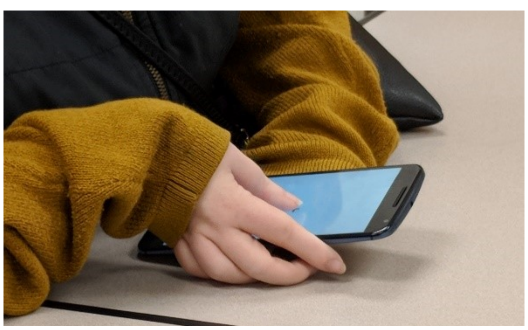

Figure 3: A participant with motor impairments completes a target selection trial on the Google Nexus 6 smartphone.

A single crosshairs was shown on the screen in each trial. Users were instructed to touch the center of the crosshairs as accurately as possible. The crosshairs were spaced equally around the screen but shown in a random order. No crosshairs's center appeared within 150 pixels of the screen border. A trial began when a finger contacted the screen and ended once the finger was lifted. A new crosshairs was displayed on the screen at the end of each trial. No feedback regarding touch locations was provided. Four (participants) × 540 (trials) = 2160 trials were collected from participants with motor impairments. We removed 54 outliers (touches that occurred more than 15 mm away from the intended target), resulting in a total of 2106 trials. Eight (participants) × 540 (trials) × 2 (postures) = 8640 trials were collected from participants without motor impairments.

3.4 Results

We collected a total of 2160 + 8640 = 10,800 trials from our 12 participants. In the following sections, we provide an analysis of our collected touch data, which focused on two properties of touch error offsets: direction and magnitude.

3.4.1 Offset Direction. Offset direction refers to the location of touches relative to the intended target. Prior research has found that the direction of offsets varies depending where on the screen users touch [9,25,55]. To analyze the direction of touch offsets, we segmented the screen into 9 regions along its *x*-axis and 15 regions along its *y*-axis. We selected these dimensions to provide an adequate number of examples per segmented region along each screen axis. We calculated the percentage of touches that occurred to the left and right of, as well above or below, the intended crosshairs in each *x*- and *y*-region. Figure 4 shows a heat map of the offset directionality along the *x*-axis (left or right), and along the *y*-axis (above or below).

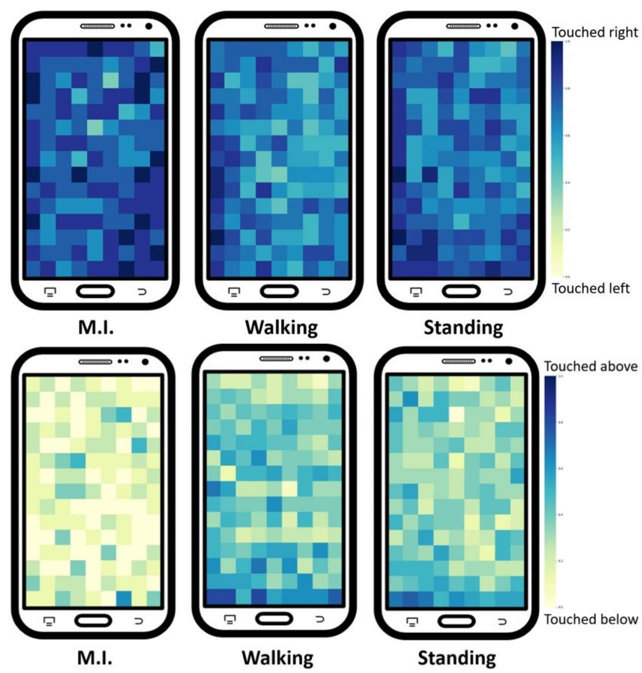

Figure 4: Heat maps of percentages of touches that occurred to the left (yellow) or right (blue) of the intended crosshairs (top), and of touches that occurred below (yellow) or above (blue) the intended crosshairs (bottom). "M.I." stands for users with motor impairments.

Participants in our three user groups exhibited similar touch behaviors across the various screen regions. As shown in the top of Figure 4, most participants' touches occurred to the right of the intended crosshairs. Overall, 83.3% of touches occurred to the right of the crosshairs for participants with motor impairments; and 63.7% and 69.7% of touches occurred to the right for users without impairments while walking and standing, respectively. Touch behaviors differ from the left to the right side of the screen, with crosshairs displayed on the right side producing a greater percentage of touches occurring to the left of the crosshairs. Across all three user groups, crosshairs shown on the left-third region of the screen resulted in 35.2% of touches occurring to the left, and crosshairs shown on the right-third of the screen resulted in 53.6% of touches occurring to the left.

Regarding the *y*-axis, most participants' touches occurred below the intended crosshairs: 84.6% of touches produced by participants with motor impairments were below the intended crosshairs, and 59.2% and 63.6% of touches occurred below for users without impairments while walking and standing, respectively. The bottom of Figure 4 shows that touches occur above the intended crosshairs more frequently as crosshairs near the bottom of the screen. For the three user groups, crosshairs shown on the top-third of the screen resulted in 31.7% of touches occurring above, and crosshairs shown on the bottom-third of the screen resulted in 42.5% of touches occurring above.

3.4.2 Offset Magnitude. The average distance between the touch and the intended crosshairs was 3.79 mm (*SD*=0.71) for participants with motor impairments, and 2.86 mm (*SD*=0.60) and 2.46 mm (*SD*=0.31) for participants without impairments while walking and standing, respectively. Similar to offset direction, the magnitude of touch errors varies depending on the screen region [22,25]. We segmented the screen into 9 regions along its *x*-axis and 15 regions along its *y*-axis and calculated the average *x*- and *y*-offset distance for each region. Figure 5 shows a heat map of the average *x*-offset (top) and *y*-offset (bottom) error.

As shown in the top half of Figure 5, the magnitude of *x*-offsets tends to be smaller on the right side of the screen compared to the left side. The average magnitude of *x*-offset errors was 2.26 mm (*SD*=0.91) for participants with motor impairments, and 1.76 mm (*SD*=0.38) and 1.53 mm (*SD*=0.25) for participants without impairments while walking and standing, respectively. Changes in magnitude of the *y*-offsets were less pronounced along the *y*-axis but accounted for more of the total touch error. The average magnitude of *y*-offset errors was 2.70 mm (*SD*=0.64) for participants with motor impairments, and 1.89 mm (*SD*=0.59) and 1.61 mm (*SD*=0.23) for people without impairments while walking and standing, respectively.

3.5 Discussion

Our results provide useful insights into the touch behaviors of people with motor impairments, and people without impairments who are walking and standing. Similar to previous investigations [25,55], our exploration of touch-offset direction found that touch locations vary depending

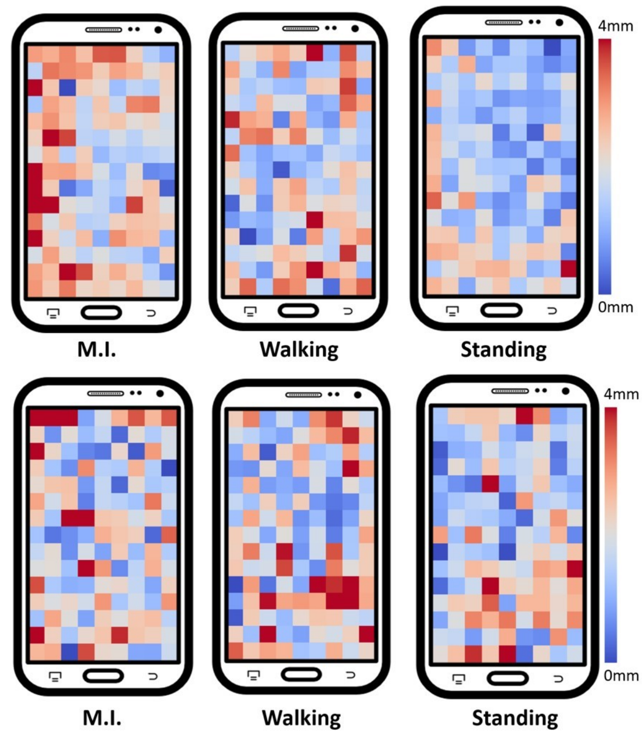

Figure 5: Heat maps of the average *x*-offset error (top) and average *y*-offset error (bottom). Areas shown in red indicate more error. "M.I." stands for users with motor impairments.

where on the screen the user is targeting. The direction of *x*- and *y*-offsets were similar across all three user groups, with participants typically aiming below and to the right of their intended crosshairs. On average, participants with motor impairments produced fewer touches above and to the left of intended crosshairs than participants who were walking or standing. However, the trend of producing more touches to the left of crosshairs located on the right portion of the screen, and touches above crosshairs located on the bottom portion of the screen, was true for all groups.

The average touch error was much higher for participants with motor impairments compared to participants walking or standing. The *y*-offset error was bigger than the x-offset error for all three user groups, but there was less noticeable change in the magnitude of y-offsets along the y-axis. The bottom heat map in Figure 5 does show higher magnitude y-offset errors along the top-left portion of the screen for participants with motor impairments, which suggests that reaching towards this region of the screen was difficult. The magnitude of x-offsets was smaller along the right side of the screen compared to the left side for the three user groups.

Taken together, these results show that the direction of touch offsets is similar for all three user groups, but the magnitude of these offsets differs across groups. Correcting erroneous offsets will allow people with motor impairments, and people undergoing the effects of situational impairments such as walking, to have more successful experiences when engaging in everyday tasks such as texting on their smartphones. Leveraging what we learned from this exploration of touch offsets, we created *Cluster Touch*, a combined user-independent and user-specific touch offset model that corrects touch offsets to provide more accurate touch predictions.

4 THE DESIGN OF CLUSTER TOUCH

Our Cluster Touch model was designed to improve the accuracy of touch input on smartphone touch screens for people with motor impairments, and for people in motor impairing situations. A key goal for Cluster Touch is to allow users to *quickly* improve the accuracy of their smartphone touch screen. Providing touch examples can require significant time and physical effort by people with motor impairments [40]. Thus, it is important that Cluster Touch improves touch accuracy while requiring as little effort from users as possible. As a result, Cluster Touch is a combined user-independent and user-specific touch model. The user-independent model captures common touch behaviors across multiple individuals. The user-independent model is then updated with touch examples provided by an individual user, allowing the model to adapt to his or her specific touch behavior. This section describes our user-independent and user-specific models.

4.1 User-Independent Model

From our exploration of touch offsets, we found that touch behaviors, such as touching to the right of crosshairs located on the left part of the screen, are similar across user groups. Cluster Touch takes advantage of this finding by combining touch examples from multiple users to create a user-independent touch model. To build our user-independent model, we collect N touch examples from individual users. Each example consists of the recorded touch location $\boldsymbol{t}$ and the location of the intended target $\boldsymbol{i}$, where $\boldsymbol{t}$ and $\boldsymbol{i}$ are two-dimensional screen coordinates. For each example, the x- and y-offsets $\boldsymbol{o_x}$ and $\boldsymbol{o_y}$ are computed by subtracting the location of the recorded touch from the location of the intended target:

$$\boldsymbol{o}_x = \boldsymbol{i}_x - \boldsymbol{t}_x$$
$$\boldsymbol{o}_y = \boldsymbol{i}_y - \boldsymbol{t}_y \qquad (1)$$

The screen is segmented into 10 equal partitions along the x- and y-axes (*i.e.*, 10 columns for the x-dimension and 10 rows for the y-dimension). Ten partitions were chosen so that each partition held approximately 10% of the data for each axis. (Note that the number of partitions used to build the user-independent model differs from the number of partitions used to explore touch offsets in the previous section.) Each x- and y-offset is placed into one of these partitions depending on the location of $\boldsymbol{i}$. The average x-offset $\boldsymbol{o}_{\overline{x}}$ is computed for each partition along the x-axis, and the average y-offset $\boldsymbol{o}_{\overline{y}}$ is computed for each partition along the y-axis:

$$\boldsymbol{o}_{\overline{x}} = \frac{\sum \boldsymbol{o}_x}{n}$$
$$\boldsymbol{o}_{\overline{y}} = \frac{\sum \boldsymbol{o}_y}{n} \qquad (2)$$

Next, we use k-means clustering [24] on our averaged binned x- and y-offsets to identify screen regions where the magnitude of touch offsets are similar along the x- and y-dimensions of the screen (see Figure 6). From multiple iterations of testing, a k of 3 was chosen, which roughly creates clusters along the top, middle, and bottom of the screen in the y-dimension, and clusters along the left, middle, and right of the screen in the x-dimension. Because k-means was computed for the binned x- and y-offsets separately, our model consists of 6 total clusters (3 along each axis). Each cluster $\boldsymbol{c}$ contains the location of the cluster in its respective x- or y-dimension, and a corrective offset. The benefit of using clusters is that it allows for the generalization of touch behaviors to screen regions, removing the need to collect touch examples from every part of the screen. As a result, to update the user-independent model, only a few examples from each cluster region are needed.

4.2 User-Specific Model

The user-independent model captures general smartphone touch behavior but is unaware of the touch abilities of any specific user. Our exploration of touch offsets showed that users' magnitudes vary between groups, with users with motor impairments exhibiting the greatest amount of error. Thus, the goal of our user-specific model is to keep the location of our found clusters—as they represent the locations where offsets tend to be similar—but to *update the magnitude of the corrective offset* of each cluster to better correct offsets produced by the given user.

To make our user-independent model user-specific, we collect touch samples from an individual user. For each sample, we calculate the x- and y-offsets (Eq. 1) and bin them in their respective dimensions according to the proximity of $\boldsymbol{i}$ to the location of each cluster $\boldsymbol{c}$. There are three bins per dimension, one for each cluster. The average x- and y-offsets are then computed for each bin (Eq. 2).

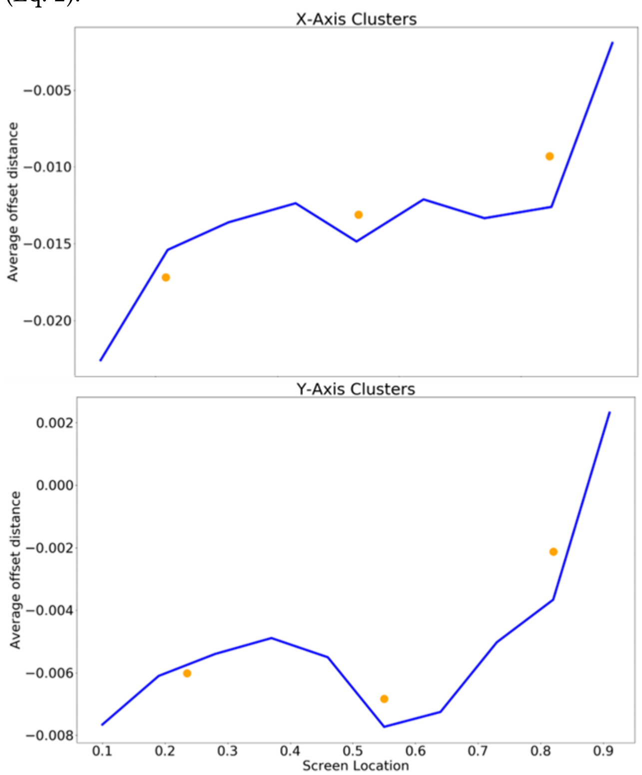

Figure 6: A plot of the average binned offsets along the screens' x- and y-axes. Cluster locations are shown in orange. Each cluster has a location (the value on the x-axes) and a corrective offset (the value on the y-axes).

Cluster offsets $\boldsymbol{c}$ are updated to $\boldsymbol{c'}$ by averaging the corrective offset for each cluster ($\boldsymbol{c}_{\overline{x}}$ and $\boldsymbol{c}_{\overline{y}}$) with the average offset ($\boldsymbol{o}_{\overline{x}}$ and $\boldsymbol{o}_{\overline{y}}$) in the bin closest to each cluster:

$$\boldsymbol{c'}_{\overline{x}} = \frac{\boldsymbol{c}_{\overline{x}} + \boldsymbol{o}_{\overline{x}}}{2}$$
$$\boldsymbol{c'}_{\overline{y}} = \frac{\boldsymbol{c}_{\overline{y}} + \boldsymbol{o}_{\overline{y}}}{2} \quad (3)$$

We tested different weighted averages and found that simply averaging the binned and cluster offsets provided additional accuracy without introducing too much variance.

4.3 Touch Prediction

Given a new touch $\boldsymbol{t}$, we find clusters $\boldsymbol{c'}_{\overline{x}}$ and $\boldsymbol{c'}_{\overline{y}}$ that are closest to $\boldsymbol{t}_x$ and $\boldsymbol{t}_y$. If $\boldsymbol{t}_x$ or $\boldsymbol{t}_y$ are located between two clusters, we compute the interpolated corrective offset between the clusters depending on the location of $\boldsymbol{t}$ relative to the two clusters. If $\boldsymbol{t}_x$ or $\boldsymbol{t}_y$ are not located between clusters (*e.g.*, a touch occurs left of the leftmost cluster), the corrective offset of the closest cluster is taken. To predict the new touch point $\boldsymbol{t'}$, we add the corrective offsets $\boldsymbol{o}$—either interpolated or taken directly from a cluster—from each dimension to the given touch:

$$\boldsymbol{t}'_x = \boldsymbol{t}_x + \boldsymbol{o}_{\overline{x}}$$
$$\boldsymbol{t}'_y = \boldsymbol{t}_y + \boldsymbol{o}_{\overline{y}} \quad (4)$$

4.4 Number of Training Examples

Because the goal of Cluster Touch is to allow users to quickly improve the accuracy of their touch screens, we tested different numbers of training examples to see how they impacted performance. We built a user-independent model from touch samples provided by participants from our exploratory study who were standing and created user-specific models with touch data from our participants with motor impairments. We incremented the number of samples E from 10 to 200 to see how well the combined model performed with various numbers of training examples (Table 1).

Table 1: Mean error distance in millimeters (lower is better) for different numbers of training examples. Standard deviations are shown in parentheses.

Mean Error Distance (mm)						
E=0	*E*=10	*E*=20	*E*=50	*E*=100	*E*=150	*E*=200
2.84 (0.59)	2.61 (0.29)	2.57 (0.24)	2.55 (0.23)	2.54 (0.23)	2.54 (0.22)	2.53 (0.22)

Table 1 shows that increasing the number of training examples can improve touch accuracy. However, this increase in accuracy comes at the expense of time, which can be problematic for people with motor impairments. For our evaluation of Cluster Touch, we decided E=20 training examples provided a good balance between speed (the time required to provide the touch samples) and accuracy.

5 EVALUATION OF CLUSTER TOUCH

To determine how accurately Cluster Touch can predict users' intended touch locations, we performed an evaluation of Cluster Touch with 12 participants with motor impairments and 12 walking participants. For our evaluation, we conducted three separate analyses. First,

				Self-reported impairments†										
ID	**Age**	**Sex**	**Health condition**	**Mo**	**Sp**	**St**	**Tr**	**Co**	**Fa**	**Gr**	**Ho**	**Se**	**Dir**	**Dis**
1	30	F	Cerebral Palsy	✓	✓	✓		✓	✓	✓	✓		✓	✓
2	21	M	Cerebral Palsy	✓	✓	✓		✓	✓	✓	✓		✓	✓
3	55	M	Multiple Sclerosis	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
4	19	F	Cerebral Palsy	✓		✓		✓	✓	✓	✓		✓	✓
5	38	F	Muscular Dystrophy	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓
6	43	F	Parkinson's Disease		✓	✓	✓	✓	✓	✓	✓		✓	✓
7	23	M	Cerebral Palsy	✓	✓	✓		✓	✓	✓	✓		✓	✓
8	39	F	Muscular Dystrophy	✓		✓		✓	✓	✓	✓		✓	✓
9	46	M	Multiple Sclerosis		✓	✓	✓	✓	✓	✓	✓		✓	✓
10	50	M	Cerebral Palsy	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
11	64	M	Parkinson's Disease	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
12	27	F	Cerebral Palsy	✓		✓		✓	✓	✓	✓	✓	✓	✓

† Mo = slow movements, Sp = spasm, Tr = tremor, Co = poor coordination, Fa = rapid fatigue, Gr = difficulty gripping, Ho = difficulty holding, Se = lack of sensation, Dir = difficulty controlling direction, Dis = difficulty controlling distance.

Table 2: Demographic information for participants with motor impairments. Self-report categories are from Findlater *et al.* [15].

we evaluated Cluster Touch's ability to predict target locations during an interactive target selection task. Second, we performed an offline crosshairs analysis of Cluster Touch and compared its performance to two machine learning-based touch offset models. Third, we performed another offline analysis to see if Cluster Touch could improve touch accuracy for actual existing mobile interfaces. For each study, we built a user-independent model using 1000 examples from participants in our exploratory study who were standing. Twenty touch examples were then used from each participant to create their combined user-independent and user-specific touch model.

5.1 Participants

We recruited 12 people with motor impairments (6 female, 6 male, average age 37.9 years, *SD*=13.7) and 12 people without impairments (7 female, 5 male, average age 24.8 years, *SD*=4.86) to participate in our study. No participants from our previous exploratory study were involved in this evaluation study. Participants were recruited through the same means as described in our exploratory study. Participants with motor impairments were paid $30 USD and participants without impairments were paid $15. All participants with motor impairments were right-handed, and 11 participants without impairments were right-handed. Additional details about our participants with motor impairments can be found in Table 2.

5.2 Apparatus

Touch data was collected using an experiment testbed developed for Android 8.1 using C# and the Xamarin framework. The testbed captured and logged all touch events. For each touch event, the reported *x*- and *y*-coordinates of the touch were captured along with its timestamp. All sessions were conducted on a Google Nexus 6 smartphone (15.14 cm diagonal with a screen resolution of 1440 × 2560 pixels) running Android 8.1.

5.3 Procedure

Each participant completed a single lab session that lasted 30 minutes to an hour. Participants with motor impairments were asked to hold the phone in whichever manner was most comfortable for them, and to touch the screen with a finger from their dominant hand. All participants with motor impairments placed the phone on the table in front of them. Participants without impairments were asked to hold the phone in their non-dominant hand and to touch the screen with a finger from their dominant hand.

Each participant completed two tasks. First, each participant completed the crosshairs selection task described above in our exploration of touch offsets. Each participant completed 540 crosshairs selection trials.

Second, participants completed a target selection task where they were instructed to select a square 6×6 mm target placed in a grid as accurately as possible. A trial started when the participant touched the screen and ended when the finger was lifted. At the end of the trial, a new square in the grid was highlighted to indicate the next target. Successfully acquired targets were briefly highlighted green and unsuccessfully acquired targets were briefly highlighted red. Participants completed this task twice, once using the touch locations generated by the phone's touch sensors and again using predictions from Cluster Touch. A user-specific model was created for each participant using only 20 touch examples collected during the crosshairs task. Examples were selected to ensure they provided good coverage of the screen. Participants were unaware that Cluster Touch was active, and no feedback about touch locations was provided in either condition. We chose not to provide location feedback to participants, as we did not want the feedback to alter participants' touch behaviors.

Participants with motor impairments completed both tasks while seated, and participants without impairments completed both tasks while standing and while walking. Twelve (participants) × 540 (trials) = 6480 crosshairs trials were completed by participants with motor impairments; and 12 (participants) × 540 (trials) × 2 (postures) = 12,960 crosshairs trials were completed by participants without motor impairments. Twelve (participants) × 50 (trials) × 2 (techniques) = 1200 interactive trials were completed by participants with motor impairments; and 12 (participants) × 50 (trials) × 2 (techniques) × 2 (postures) = 2400 interactive trials were completed by participants without motor impairments.

5.4 Design and Analysis

We completed separate analyses for data collected from participants with and without motor impairments. We conducted three experiments with each group.

The first was to determine if Cluster Touch could improve touch accuracy over the native touch sensor during an interactive target selection task. For our analysis of participants with motor impairments, our experiment was a within-subjects design with one factor of two levels:

- *Technique*: Nexus, Cluster Touch

Our experiment for participants without motor impairments was a 2×2 within-subjects design with the same *Technique* factor as above, but with an additional factor of two levels:

- *Posture*: Standing, Walking

In all experiments, "*Nexus*" refers to the touch-down location reported by the native touch sensor on the Google Nexus 6. The presentation of *Technique* was counterbalanced to account for order effects.

Our second experiment compared target prediction accuracy of Cluster Touch to the native touch sensor in the Nexus, and to two machine learning-based touch-offset models previously shown to improve touch accuracy [9,55]. Our analysis for participants with motor impairments was a within-subjects design with one factor of four levels:

- *Technique*: Nexus, Cluster Touch, Gaussian Process [55], Linear Offset [9]

The experiment for participants without motor impairments was a 4×2 within-subjects design with the same *Technique* factor as above and the same *Posture* factor as in the first experiment (standing, walking).

For this analysis, we implemented a testing procedure based on machine learning evaluations [35]. For each participant, we randomly selected 20 touch samples (*i.e.*, trials from the crosshairs selection task) as training examples for the three models in *Technique*. We used these models to predict intended touch points for the remaining trials. This procedure was performed 100 times for each participant, with a new set of training examples being randomly selected each time. We used the recommended parameters described in prior work [7] for the Gaussian Process [55] and Linear Offset [9] models. To train the models, we used the 1000 examples used to train the user-independent model as well the 20 touch samples used to train the user-specific portion of the combined model.

For our third experiment, we conducted an offline analysis of actual mobile applications to determine whether Cluster Touch could improve touch accuracy over the native touch sensor. We extracted 15,231 clickable targets from mobile interfaces in the RICO dataset [12]. For each crosshairs trial, we checked whether the location of the crosshairs for that trial was within the bounds of any target. Next, we used Cluster Touch to predict a new touch location for that trial. A "hit" occurred if the predicted touch landed within the bounds of the target. Our analysis for participants with motor impairments was a within-subjects design with the same *Technique* factor as in the first experiment.

For participants without motor impairments, our experiment was a 2×2 within-subjects design with the same *Technique* factor as above and the same *Posture* factor as in the experiments above (standing, walking).

Target *Hit Rate*, the proportion of touches that successfully fell within the bounds of the target, was the dependent variable in the first and third experiments. The dependent variable of interest for our second experiment was *Error Distance*, measured by the Euclidean distance between the center of the crosshairs in each trial and the predicted target location of each technique. A mixed-effects model analysis of variance [34] was used to analyze *Error Distance*. Our model used fixed-effects for *Technique* and *Posture*. *Subject* was a random effect to accommodate repeated measures. *Post hoc* pairwise comparisons were computed using Holm's sequential Bonferroni procedure [27] to correct for multiple comparisons. The nonparametric aligned rank transform procedure [26,45,56] was used to analyze *Hit Rate* for participants without motor impairments, as this measure did not conform to the assumptions of ANOVA. Nonparametric Wilcoxon signed-rank tests were used to analyze *Hit Rate* for participants with motor impairments.

6 RESULTS

This section presents the results of our three experiments to determine the effectiveness of Cluster Touch for participants with and without motor impairments.

6.1 Interactive Selection Task

For participants with motor impairments, the average *Hit Rate* was 85.52% (*SD*=3.06%) with Cluster Touch and 74.50% (*SD*=3.88%) for Nexus. This difference was statistically significant (Z=-39.00, p<.001). The average *Hit Rates* for participants without motor impairments are shown in Table 3. There was a significant main effect of *Technique* ($F_{1,33}$=73.69, p<.0001) and *Posture* ($F_{1,33}$=58.05, p<.0001) on *Hit Rate*. There was no significant *Technique* × *Posture* interaction ($F_{1,33}$=0.63, *n.s.*).

Table 3: Overall means for *Hit Rate* (higher is better) for levels of *Technique* and *Posture* for participants without motor impairments. Standard deviations are in parentheses.

Mean Hit Rate (%)

Technique	Posture	Hit Rate
Cluster Touch	Standing	95.33% (1.97%)
Nexus	Standing	91.67% (2.67%)
Cluster Touch	Walking	92.04% (2.07%)
Nexus	Walking	86.33% (2.90%)

6.2 Crosshairs Prediction

Table 4 shows mean *Error Distance* for participants with and without motor impairments. There was a significant effect of *Technique* on *Error Distance* ($F_{3,33}$=109.34, p<.0001) for our participants with motor impairments. Pairwise comparisons showed that Cluster Touch was more accurate than Nexus, Gaussian Process, and Linear Offset (p<.01). For our group without impairments, there was a significant effect of *Technique* ($F_{3,77}$=7.57, p<.001) and *Posture* ($F_{3,77}$=168.93, p<.0001) on *Error Distance*. There was no significant *Technique* × *Posture* interaction ($F_{3,77}$=0.58, *n.s.*). Pairwise comparisons showed that Cluster Touch was significantly more accurate than Nexus (p<.001).

Table 4: Overall means for *Error Distance* (lower is better) for levels of *Technique* and *Posture* for participants with and without motor impairments. Standard deviations are in parentheses.

Mean Error Distance (mm)

Group	Cluster Touch	Gaussian Process	Linear Offset	Google Nexus 6
M.I.	2.81 (0.28)	3.17 (0.34)	3.15 (0.34)	3.42 (0.37)
Standing	2.22 (0.24)	2.31 (0.22)	2.31 (0.22)	2.41 (0.26)
Walking	2.67 (0.34)	2.80 (0.29)	2.80 (0.29)	2.85 (0.30)

6.3 RICO UI Analysis

For participants with motor impairments, the average *Hit Rate* was 90.04% (*SD*=3.30%) for Cluster Touch and 82.97% (*SD*=4.80%) for Nexus. This difference was statistically significant (Z=-39.00, p<.001). Average *Hit Rates* for participants without motor impairments are shown in Table 5. There was a significant main effect of *Technique* ($F_{1,33}$=82.45, p<.001) and *Posture* ($F_{1,33}$=72.29, p<.0001) on *Hit Rate*. There was no significant *Technique* × *Posture* interaction ($F_{1,33}$=3.67, *n.s.*).

Table 5: Overall means for *Hit Rate* (higher is better) for levels of *Technique* and *Posture*. Standard deviations are shown in parentheses.

Mean Hit Rate (%)

Technique	Posture	Hit Rate
Cluster Touch	Standing	95.23% (1.67%)
Nexus	Standing	93.02% (2.01%)
Cluster Touch	Walking	93.22% (1.85%)
Nexus	Walking	89.08% (2.27%)

7 DISCUSSION

The results from our experiment show that Cluster Touch can significantly improve smartphone touch screen accuracy for people with motor impairments and people in motor-impairing situations like walking. In the interactive selection tasks, Cluster Touch was 14.65% more accurate than the touch-down location reported by the Nexus 6's native sensor. These results are encouraging, as they demonstrate that Cluster Touch can be used in real time to improve smartphone touch accuracy. Furthermore, this increase in accuracy only required participants to provide 20 touch examples. Reducing the time and effort required to calibrate smartphone touch screens is an important goal, as users with motor impairments might struggle to complete a laborious calibration procedure. Our offline analysis comparing Cluster Touch to the Gaussian Process [55] and Linear Offset [9] models showed that Cluster Touch can improve touch accuracy just as well or better than these touch models based on machine learning when the number of user-provided examples is small. For participants with motor impairments, Cluster Touch was 11.36% more accurate than the Gaussian Process model and 10.79% more accurate than the Linear Offset model. Cluster Touch benefits from its combined user-independent and user-specific touch models, as a greater percentage of user touch behavior is represented compared to shared models based on machine learning.

Cluster Touch was, on average, more accurate for users while walking compared to the Nexus in the first experiment, increasing touch accuracy by 6.81%. Cluster Touch was also slightly more accurate in predicting target locations for walking participants than the Gaussian Process [55] and Linear Offset [9] models, improving accuracy by 4.64% over both models. These results demonstrate that Cluster Touch could also provide benefits to users experiencing the effects of situational impairments like walking. Smartphones could detect when a user begins to walk and switch to the walking model, then switch back to the standing model when the user stops walking. Cluster Touch also improved average touch accuracy for participants while standing by 4.37%, which suggests that this model could be used by a wide assortment of users or applications. For example, mobile game developers might implement Cluster Touch to improve touch accuracy while gaming.

Our analysis of existing targets in the RICO dataset [12] showed that Cluster Touch can improve touch accuracy for various real mobile applications. Of the more than 15,000 interface targets we tested, Cluster Touch improved touch accuracy by 8.21% over the Nexus 6 for participants with motor impairments. Improved touch accuracy of mobile applications could be a huge benefit to people with motor impairments, as more recreational and work-related applications transition from the desktop to smartphones.

Because touch locations were reported on touch-down, we did not record slippage in our studies. Prior research has shown that slippage can occur when users with motor impairments interact with touch-enabled devices [36,51]. Touch models for people with motor impairments can leverage designs intended to help with slippage, such as the *Steadied-bubbles* technique [36], which locks the location of the touch (or pen) on touch-down.

7.1 Limitations and Future Work

A limitation of this work is that we only trained touch models for one pose, namely holding the phone in the non-dominant hand and touching the screen with the index finger from the dominant hand. Also, all but one of our study participants were right-handed, meaning that some of the touch behaviors we identified could be different for left-handed users. If so, Cluster Touch could build different user-independent models for left-handed users and for different phone grips. We could then use techniques that can identify changes in grip and handedness [21,43] to load the appropriate model. Further analysis of how different grips, such as one-handed use, impact model creation and performance is left for future work.

Another limitation is that our study participants were experienced smartphone users, which means they might be more accurate than novice smartphone users, or than people for whom touch screens pose significant accessibility challenges [40,51]. Further research is needed to collect touch data from more participants with motor impairments (*i.e.*, hundreds instead of dozens) to understand how touch screen experience and motor capabilities impact the creation and accuracy of our models.

Other future work includes understanding how Cluster Touch might accommodate changes in touch behavior over time, as the touch abilities of people with motor impairments might fluctuate throughout the day for various reasons, such as fatigue or the effects of medication. It is also important to understand how Cluster Touch performs on other smartphones, and to understand if a Cluster Touch model created on one smartphone can transfer effectively to a different smartphone.

8 CONCLUSION

We have investigated touch offsets created by people with motor impairments, and by people without impairments while walking and standing. Our analysis found that users across groups exhibited similar touch behaviors regarding where touches occur relative to their intended targets, but that the magnitude of these errors was more pronounced for people with motor impairments and for walking people than for people who were standing. To improve touch screen accuracy for people with motor impairments, and for people in motor-impairing situations, we created Cluster Touch, a combined user-independent and user-specific touch model that can improve touch accuracy by collecting only 20 touch examples from the target user. In an evaluation of Cluster Touch, we found that it was significantly more accurate in predicting intended target locations for people with motor impairments than the touch sensors found in the Google Nexus 6 smartphone. Cluster Touch also improved accuracy over two prior statistical machine learning models, Gaussian Process [55] and Linear Offset [9]. We also found that Cluster Touch was able to improve touch accuracy for people without impairments who were walking, demonstrating that Cluster Touch has the potential to provide accuracy improvements for a range of users in different situations.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under grant IIS-1217627, by a Google Faculty Award, and by a Microsoft Research Dissertation Grant. Any opinions, findings, conclusions or recommendations expressed in our work are those of the authors and do not necessarily reflect those of any supporter.

REFERENCES

[1] Lisa Anthony, YooJin Kim, and Leah Findlater. 2013. Analyzing user-generated Youtube videos to understand touchscreen use by people with motor impairments. *Proceedings of ACM CHI 2013*, 1223–1232. https://doi.org/10.1145/2470654.2466158

[2] Ravin Balakrishnan. 2004. “Beating” Fitts’ law: Virtual enhancements for pointing facilitation. *International Journal of Human-Computer Studies* 61, 6: 857–874. https://doi.org/10.1016/j.ijhcs.2004.09.002

[3] Joanna Bergstrom-Lehtovirta, Antti Oulasvirta, and Stephen Brewster. 2011. The effects of walking speed on target acquisition on a touchscreen interface. *Proceedings of ACM MobileHCI 2011*, 143–146. https://doi.org/10.1145/2037373.2037396

[4] Xiaojun Bi, Yang Li, and Shumin Zhai. 2013. FFitts law: Modeling finger touch with Fitts’ law. *Proceedings of ACM CHI 2013*, 1363–1372. https://doi.org/10.1145/2470654.2466180

[5] Xiaojun Bi and Shumin Zhai. 2013. Bayesian touch: A statistical criterion of target selection with finger touch. *Proceedings of ACM UIST 2013*, 51–60. https://doi.org/10.1145/2501988.2502058

[6] Pradipta Biswas and Patrick Langdon. 2012. Developing multimodal adaptation algorithm for mobility impaired users by evaluating their hand strength. *International Journal of Human-Computer Interaction* 28, 9: 576–596. https://doi.org/10.1080/10447318.2011.636294

[7] Daniel Buschek and Florian Alt. 2015. TouchML: A Machine learning toolkit for modelling spatial touch targeting behaviour. *Proceedings of ACM IUI 2015*, 110–114. https://doi.org/10.1145/2678025.2701381

[8] Daniel Buschek and Florian Alt. 2017. ProbUI: Generalising touch target representations to enable declarative gesture definition for probabilistic GUIs. *Proceedings of ACM CHI 2017*, 4640–4653. https://doi.org/10.1145/3025453.3025502

[9] Daniel Buschek, Simon Rogers, and Roderick Murray-Smith. 2013. User-specific touch models in a cross-device context. *Proceedings of ACM MobileHCI 2013*, 382–391. https://doi.org/10.1145/2493190.2493206

[10] Karen B. Chen, Anne B. Savage, Amrish O. Chourasia, Douglas A. Wiegmann, and Mary E. Sesto. 2013. Touch screen performance by individuals with and without motor control disabilities. *Applied Ergonomics* 44, 2: 297–302. https://doi.org/10.1016/j.apergo.2012.08.004

[11] Andrew Crossan, Roderick Murray-Smith, Stephen Brewster, James Kelly, and Bojan Musizza. 2005. Gait phase effects in mobile interaction. *Extended Abstracts of CHI 2005*, 1312–1315. https://doi.org/10.1145/1056808.1056904

[12] Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. 2017. RICO: A mobile app dataset for building data-driven design applications. *Proceedings of ACM UIST 2017*, 845–854. https://doi.org/10.1145/3126594.3126651

[13] Morgan Dixon, James Fogarty, and Jacob O. Wobbrock. 2012. A general-purpose target-aware pointing enhancement using pixel-level analysis of graphical interfaces. *Proceedings of ACM CHI 2012*, 3167–3176. https://doi.org/10.1145/2207676.2208734

[14] Sacha N. Duff, Curt B. Irwin, Jennifer L. Skye, Mary E. Sesto, and Douglas A. Wiegmann. 2010. The effect of disability and approach on touch screen performance during a number entry task. *Proceedings of the Human Factors and Ergonomics Society Annual Conference (HFES ’10)*, 566–570.

[15] Leah Findlater, Alex Jansen, Kristen Shinohara, Morgan Dixon, Peter Kamb, Joshua Rakita, and Jacob O. Wobbrock. 2010. Enhanced area cursors: Reducing fine pointing demands for people with motor impairments. *Proceedings of ACM UIST 2010*, 153–162. https://doi.org/10.1145/1866029.1866055

[16] Leah Findlater, Karyn Moffatt, Jon E. Froehlich, Meethu Malu, and Joan Zhang. 2017. Comparing touchscreen and mouse input performance by people with and without upper body motor impairments. *Proceedings of ACM CHI 2017*, 6056–6061. https://doi.org/10.1145/3025453.3025603

[17] Jon Froehlich, Jacob O. Wobbrock, and Shaun K. Kane. 2007. Barrier pointing: Using physical edges to assist target acquisition on mobile device touch screens. *Proceedings of ACM ASSETS 2007*, 19–26. https://doi.org/10.1145/1296843.1296849

[18] Krzysztof Gajos, Jacob O. Wobbrock, and Daniel S. Weld. 2007. Automatically generating user interfaces adapted to users' motor and vision capabilities. *Proceedings of ACM UIST 2007*, 231–240. https://doi.org/10.1145/1294211.1294253

[19] Krzysztof Z. Gajos, Jacob O. Wobbrock, and Daniel S. Weld. 2008. Improving the performance of motor-impaired users with automatically-generated, ability-based interfaces. *Proceedings of ACM CHI 2008*, 1257–1266. https://doi.org/10.1145/1357054.1357250

[20] Mayank Goel, Leah Findlater, and Jacob O. Wobbrock. 2012. WalkType: Using accelerometer data to accomodate situational impairments in mobile touch screen text entry. *Proceedings of ACM CHI 2012*, 2687–2696. https://doi.org/10.1145/2207676.2208662

[21] Mayank Goel, Jacob O. Wobbrock, and Shwetak Patel. 2012. GripSense: Using built-in sensors to detect hand posture and pressure on commodity mobile phones. *Proceedings of ACM UIST 2012*, 545–554. https://doi.org/10.1145/2380116.2380184

[22] Tiago Guerreiro, Hugo Nicolau, Joaquim Jorge, and Daniel Gonçalves. 2010. Assessing mobile touch interfaces for tetraplegics. *Proceedings of ACM MobileHCI 2010*, 31–34. https://doi.org/10.1145/1851600.1851608

[23] Tiago Guerreiro, Hugo Nicolau, Joaquim Jorge, and Daniel Gonçalves. 2010. Towards accessible touch interfaces. *Proceedings of ACM ASSETS 2010*, 19–26. https://doi.org/10.1145/1878803.1878809

[24] J. A. Hartigan and M. A. Wong. 1979. Algorithm AS 136: A K-Means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28, 1: 100–108. https://doi.org/10.2307/2346830

[25] Niels Henze, Enrico Rukzio, and Susanne Boll. 2011. 100,000,000 Taps: Analysis and improvement of touch performance in the large. *Proceedings of ACM MobileHCI 2011*, 133–142. https://doi.org/10.1145/2037373.2037395

[26] James J. Higgins and S Tashtoush. 1994. An aligned rank transform test for interaction. *Nonlinear World* 1, 2: 201–211.

[27] Sture Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics* 6, 2: 65–70.

[28] Christian Holz and Patrick Baudisch. 2010. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. *Proceedings of ACM CHI 2010*, 581–590. https://doi.org/10.1145/1753326.1753413

[29] Christian Holz and Patrick Baudisch. 2011. Understanding touch. *Proceedings of ACM CHI 2011*, 2501–2510. https://doi.org/10.1145/1978942.1979308

[30] Curt B. Irwin and Mary E. Sesto. 2012. Performance and touch characteristics of disabled and non-disabled participants during a reciprocal tapping task using touch screen technology. *Applied Ergonomics* 43, 6: 1038–1043. https://doi.org/10.1016/j.apergo.2012.03.003

[31] Shaun K. Kane, Chandrika Jayant, Jacob O. Wobbrock, and Richard E. Ladner. 2009. Freedom to roam: A study of mobile device adoption and accessibility for people with visual and motor disabilities. *Proceedings of ACM ASSETS 2009*, 115–122. https://doi.org/10.1145/1639642.1639663

[32] Shaun K. Kane, Jacob O. Wobbrock, and Ian E. Smith. 2008. Getting off the treadmill: Evaluating walking user interfaces for mobile devices in public spaces. *Proceedings of ACM MobileHCI 2008*, 109–118. https://doi.org/10.1145/1409240.1409253

[33] Min Lin, Rich Goldman, Kathleen J. Price, Andrew Sears, and Julie Jacko. 2007. How do people tap when walking? An empirical investigation of nomadic data entry. *International Journal of Human-Computer Studies* 65, 9: 759–769. https://doi.org/10.1016/j.ijhcs.2007.04.001

[34] R. C. Littell, P. R. Henry, and C. B. Ammerman. 1998. Statistical analysis of repeated measures data using SAS procedures. *Journal of Animal Science* 76, 4: 1216–1231.

[35] Tom M. Mitchell. 1997. *Machine Learning*. McGraw-Hill Education, New York.

[36] Karyn Moffatt and Joanna McGrenere. 2010. Steadied-bubbles: Combining techniques to address pen-based pointing errors for younger and older adults. *Proceedings of ACM CHI 2010*, 1125–1134. https://doi.org/10.1145/1753326.1753495

[37] Kyle Montague, Vicki L. Hanson, and Andy Cobley. 2012. Designing for individuals: Usable touch-screen interaction through shared user models. *Proceedings of ACM ASSETS 2012*, 151–158. https://doi.org/10.1145/2384916.2384943

[38] Kyle Montague, Hugo Nicolau, and Vicki L. Hanson. 2014. Motor-impaired touchscreen interactions in the wild. *Proceedings of ACM ASSETS 2014*, 123–130. https://doi.org/10.1145/2661334.2661362

[39] Martez E. Mott, Jane E., Cynthia L. Bennett, Edward Cutrell, and Meredith Ringel Morris. 2018. Understanding the accessibility of smartphone photography for people with motor impairments. *Proceedings of ACM CHI 2018*, 520:1–520:12. https://doi.org/10.1145/3173574.3174094

[40] Martez E. Mott, Radu-Daniel Vatavu, Shaun K. Kane, and Jacob O. Wobbrock. 2016. Smart Touch: Improving touch accuracy for people with motor Impairments with template matching. *Proceedings of ACM CHI 2016*, 1934–1946. https://doi.org/10.1145/2858036.2858390

[41] Josip Musić, Daryl Weir, Roderick Murray-Smith, and Simon Rogers. 2016. Modelling and correcting for the impact of the gait cycle on touch screen typing accuracy. *mUX: The Journal of Mobile User Experience* 5, 1: 1–11. https://doi.org/10.1186/s13678-016-0002-3

[42] Maia Naftali and Leah Findlater. 2014. Accessibility in context: Understanding the truly mobile experience of smartphone users with motor impairments. *Proceedings of ACM ASSETS 2014*, 209–216. https://doi.org/10.1145/2661334.2661372

[43] Matei Negulescu and Joanna McGrenere. 2015. Grip change as an information side channel for mobile touch interaction. *Proceedings of ACM CHI 2015*, 1519–1522. https://doi.org/10.1145/2702123.2702185

[44] Alexander Ng. 2014. The effects of encumbrance on mobile interactions. *Proceedings of ACM MobileHCI 2014*, 405–406. https://doi.org/10.1145/2628363.2634268

[45] K. C. Salter and R. F. Fawcett. 1993. The art test of interaction: a robust and powerful rank test of interaction in factorial models. *Communications in Statistics - Simulation and Computation* 22, 1: 137–153. https://doi.org/10.1080/03610919308813085

[46] Julia Schwarz, Scott Hudson, Jennifer Mankoff, and Andrew D. Wilson. 2010. A framework for robust and flexible handling of inputs with uncertainty. *Proceedings of ACM UIST 2010*, 47–56. https://doi.org/10.1145/1866029.1866039

[47] Julia Schwarz, Jennifer Mankoff, and Scott Hudson. 2011. Monte Carlo methods for managing interactive state, action and feedback under uncertainty. *Proceedings of ACM UIST 2011*, 235–244. https://doi.org/10.1145/2047196.2047227

[48] Julia Schwarz, Jennifer Mankoff, and Scott E. Hudson. 2015. An architecture for generating interactive feedback in probabilistic user interfaces. *Proceedings of ACM CHI 2015*, 2545–2554. https://doi.org/10.1145/2702123.2702228

[49] Andrew Sears, Min Lin, Julie Jacko, and Yan Xiao. 2003. When computers fade… Pervasive computing and situationally-induced impairments and disabilities. *Proceedings of HCI International 2003*, 1298–1302.

[50] Andrew Sears and Mark Young. 2003. Physical disabilities and computing technologies: an analysis of impairments. In Julie A. Jacko and Andrew Sears (eds.). Lawrence Erlbaum, Hillsdale, NJ, 482–503. http://dl.acm.org/citation.cfm?id=772072.772105

[51] Shari Trewin, Cal Swart, and Donna Pettick. 2013. Physical accessibility of touchscreen smartphones. *Proceedings of ACM ASSETS 2013*, 19:1–19:8. https://doi.org/10.1145/2513383.2513446

[52] Daniel Vogel and Patrick Baudisch. 2007. Shift: A technique for operating pen-based interfaces using touch. *Proceedings of ACM CHI 2007*, 657–666. https://doi.org/10.1145/1240624.1240727

[53] Chat Wacharamanotham, Jan Hurtmanns, Alexander Mertens, Martin Kronenbuerger, Christopher Schlick, and Jan Borchers. 2011. Evaluating swabbing: A touchscreen input method for elderly users with tremor. *Proceedings of ACM CHI 2011*, 623–626. https://doi.org/10.1145/1978942.1979031

[54] Daryl Weir. 2012. Machine learning models for uncertain interaction. *Adjunct Proceedings of ACM UIST 2012*, 31–34. https://doi.org/10.1145/2380296.2380313

[55] Daryl Weir, Simon Rogers, Roderick Murray-Smith, and Markus Löchtefeld. 2012. A user-specific machine learning approach for improving touch accuracy on mobile devices. *Proceedings of ACM UIST 2012*, 465–476. https://doi.org/10.1145/2380116.2380175

[56] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011. The Aligned Rank Transform for nonparametric factorial analyses using only ANOVA procedures. *Proceedings of ACM CHI 2011*, 143–146. https://doi.org/10.1145/1978942.1978963

[57] Jacob O. Wobbrock, Krzysztof Z. Gajos, Shaun K. Kane, and Gregg C. Vanderheiden. 2018. Ability-based design. *Communications of the ACM* 61, 6: 62–71. https://doi.org/10.1145/3148051

[58] Jacob O. Wobbrock, Shaun K. Kane, Krzysztof Z. Gajos, Susumu Harada, and Jon Froehlich. 2011. Ability-based design: Concept, principles and examples. *ACM Transactions on Accessible Computing* 3, 3: 9:1–9:27. https://doi.org/10.1145/1952383.1952384

[59] Jacob O. Wobbrock, Brad A. Myers, and John A. Kembel. 2003. EdgeWrite: A stylus-based text entry method designed for high accuracy and stability of motion. *Proceedings of ACM UIST 2003*, 61–70. https://doi.org/10.1145/964696.964703