

# The Generalized Perceived Input Point Model and How to Double Touch Accuracy by Extracting Fingerprints

Christian Holz and Patrick Baudisch

Hasso Plattner Institute

Potsdam, Germany

{christian.holz, patrick.baudisch}@hpi.uni-potsdam.de

## ABSTRACT

It is generally assumed that touch input cannot be accurate because of the *fat finger problem*, i.e., the softness of the fingertip combined with the occlusion of the target by the finger. In this paper, we show that this is not the case. We base our argument on a new model of touch inaccuracy. Our model is not based on the fat finger problem, but on the *perceived input point* model. In its published form, this model states that touch screens report touch location at an offset from the intended target. We generalize this model so that it represents offsets for individual finger postures and users. We thereby switch from the traditional 2D model of touch to a model that considers touch a phenomenon in 3-space. We report a user study, in which the generalized model explained 67% of the touch inaccuracy that was previously attributed to the fat finger problem.

In the second half of this paper, we present two devices that exploit the new model in order to improve touch accuracy. Both model touch on per-posture and per-user basis in order to increase accuracy by applying respective offsets. Our *RidgePad* prototype extracts posture and user ID from the user's fingerprint during each touch interaction. In a user study, it achieved 1.8 times higher accuracy than a simulated capacitive baseline condition. A prototype based on optical tracking achieved even 3.3 times higher accuracy. The increase in accuracy can be used to make touch interfaces more reliable, to pack up to  $3.3^2 > 10$  times more controls into the same surface, or to bring touch input to very small mobile devices.

## Author Keywords

Touch, touch pad, touch screen, precision, targeting, fingerprint scanner, 6DOF, mobile devices, input, pointing.

## ACM Classification Keywords

H5.2 [Information interfaces and presentation]: User Interfaces. Input devices & strategies; B4.2 Input Output devices.

## General Terms

Design, Experimentation, Human Factors, Measurement, Performance, Theory.

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

CHI 2010, April 10–15, 2010, Atlanta, Georgia, USA.

Copyright 2010 ACM 978-1-60558-929-9/10/04...\$10.00.

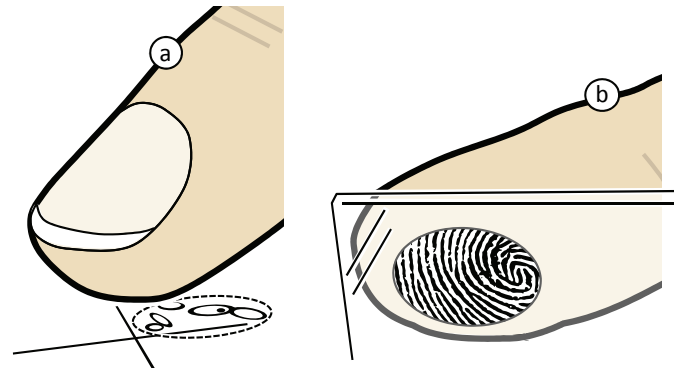


Figure 1: (a) The Generalized Perceived Input Point Model: a user has repeatedly acquired the shown crosshairs using finger postures ranging from 90° (vertical) to 15° pitch (almost horizontal). The five white ovals each contain 65% of the resulting contact points. The key observation is that the ovals are offset with respect to each other, yet small. We find a similar effect across different levels of finger roll and finger yaw, and across users. We conclude that the inaccuracy of touch (dashed oval) is primarily the result of failure to distinguish between different users and finger postures, rather than the *fat finger problem* [18]. (b) The ridges of this fingerprint belong to the front region of a fingertip. Our *RidgePad* prototype uses this observation to deduce finger posture and user ID during each touch. This allows it to exploit the new model and obtain 1.8 times higher accuracy than capacitive sensing.

## INTRODUCTION

Acquiring a small target on a touch screen is error prone. We can examine how inaccurate touch is on a given device by letting users acquire a small target repeatedly: the more inaccurate the device, the wider spread the distribution of the sensed contact points (indicated with a dashed outline in Figure 1a; the user was targeting the crosshairs). When acquiring a target of finite size, such as a button, wider spread results in an increased risk of missing the target.

The common explanation for the inaccuracy of touch is the *fat finger problem* [18]. In this model, the softness of the user's skin causes the touch position to be sensed anywhere within the contact area between the user's fingertip and the device. At the same time, the finger occludes the target. This prevents the target from providing visual feedback so that users cannot compensate for the randomness.

Researchers have therefore argued that users *cannot* reliably acquire targets smaller than a certain size. Minimum target sizes have been reported as anywhere between

10.5mm [18] and 26mm per side [11]. Larger targets, however, require large screens, which is especially problematic on mobile devices, where screen space is scarce [3].

### **AN ALTERNATIVE MODEL FOR THE INACCURACY OF TOUCH: PERCEIVED INPUT POINT, NOT FAT FINGERS**

While the fat finger problem has received a lot of attention, we argue that it is *not* the true reason for the inaccuracy of touch. We argue that the *perceived input point model* is the primary source of the problem; Vogel and Baudisch mention it in the same paper that discusses the fat finger problem [18]. When a user tries to acquire a target, the center of the contact area tends to be located a couple of millimeters off the target location—typically “below” the target (the black dot in Figure 1a). The fact that touch devices report the touch location at an offset increases the risk of missing the target. Unlike the fat finger problem, however, the “perceived input point” is a *systematic* effect. This allows compensating for the effect by applying an inverse offset when recognizing touch [18].

In this paper, we generalize the perceived input point model in order to reduce touch inaccuracy even further. We hypothesize that the offset between the center of the contact area and the target depends not only on the *x/y* location of the target, but also on the *wider context* of the touch interaction. The *wider context* in this *generalized perceived input point model* could potentially include a larger number of variables. In this paper, we examine the following four:

**1-3. Angle** between the finger and the touch surface (roll, pitch, and yaw). The related work suggests that pointing might be affected by changes in finger orientation (also called ‘yaw’) [20] and finger “steepness” (or ‘pitch’) [9]. We also include roll. By considering all three finger angles we implicitly switch from the traditional 2D model of touch to a model that considers touch a phenomenon in 3-space.

**4. User.** Each user might have a different mental model of how to acquire the target. While touch is well understood in the macroscopic world (most people will agree on whether a person is touching the seat or the backrest of a chair), note that there is probably no universally agreed upon interpretation for determining what *exact* location a finger is touching.

To verify these assumptions we conducted a user study. Figure 1a gives a preview of our findings. The oval chart is taken from one of the participants of the study who repeatedly acquired a crosshair target with five different levels of finger pitch ranging from 15° to 90°; each of the five white ovals shown in Figure 1 contains 65% of one of the resulting contact point distributions. The key observation is that all five ovals are *offset* with respect to each other, yet small. We find a similar effect across different levels of finger roll and finger yaw, and across users. We conclude that the inaccuracy of touch (Figure 1a, dashed outline) is primarily the result of failure to distinguish between different users and finger postures, rather than the fat finger problem [18]. We consider this as support for the proposed model.

In the second half of this paper, we switch to an engineering perspective. We present two devices that exploit the new model in order to achieve higher touch accuracy. Both devices model touch on per-posture and per-user basis and increase accuracy by applying respective offsets. Figure 1b illustrates the workings of one of them, the *RidgePad* prototype. It extracts user ID and posture from the user’s fingerprint during each touch interaction. The ridges of the shown fingerprint belong to the front region of a fingertip. *RidgePad* uses this observation to deduce finger posture. This allows *RidgePad* to exploit the new model and obtain 1.8x higher accuracy than capacitive sensing. A prototype based on optical tracking achieved even 3.3 times higher accuracy.

The increase in accuracy can be used to make touch interfaces more reliable, pack up to  $3.3^2 > 10$  times more controls into the same surface, or, assuming future miniaturization, to bring touch input to very small mobile devices.

### **RELATED WORK**

The work presented in this paper is related to targeting aids, impact of finger angles on touch, and fingerprint recognition.

#### **Targeting aids**

A popular approach to precise touch is to address the occlusion problem using targeting aids. Zooming-based techniques reduce occlusion (e.g., [2], *TapTap* [15]) although they cannot fully resolve it. Other targeting aids remove occlusion entirely by separating the user’s hand from the pointer (e.g., *styli* [14], *offset cursor* [13], *shift* [18], *cross-lever* and *precision handle* [2], and *back-of-device interaction* [3]).

On the flipside, targeting aids make touch less direct and may therefore reduce the intuitiveness of input. They also increase targeting time; *offset cursor*, for example, incurs a task time penalty of 250ms to 1000ms [18].

#### **Sensing Finger Angles and Applications**

Different types of touch technologies are able to extract different subsets of finger posture. Microsoft Surface detects finger orientation by analyzing the diffuse reflection of the hovering hand. Capacitive technologies, such as the *FingerWorks iGesture* pad [8] estimate finger orientation based on the eccentricity of the contact area.

Some researchers have proposed exploiting finger postures in order to enable additional functionality. Wang and Ren, for example, proposed occlusion-free pie menus [20] and hand orientation-aware gesture processing [19]. Their algorithm detects finger yaw by observing the changes in contact area over time.

Finger *roll* has been proposed as the basis for a new gesture language (*MicroRolls* [16]). The specific implementation of *MicroRolls*, however, cannot distinguish between finger rolling and finger dragging. Therefore rolling serves here primarily as an alternative way of performing a drag gesture. In contrast, the *RidgePad* prototype presented in this

paper can distinguish rolling from dragging (Figure 9). It can also extract rolling and dragging from a single interaction.

Only little research has been done on the impact of finger posture on touch. Forlines et al. found that acquiring a target on the opposite side of a tabletop system results in a flatter finger pitch and thus a larger contact area [9]. Wang and Ren examined the impact of specific finger postures and gestures on input accuracy and contact area [20].

### Minimum Target Size

It is commonly understood that a target needs to have a certain minimum size in order to allow for reliable acquisition. However, different authors find different values for this minimum target size, e.g., 26mm (Hall et al. [11]), 11.5mm (Wang and Ren [20]), and 10.5mm (Vogel and Baudisch [18]).

In the light of the generalized perceived input point model, it seems plausible that the disagreement about minimum button sizes was caused by differences in study conditions,

Pitch, for example, has the two levels ‘fingertip’ vs. ‘nail’ for Vogel and Baudisch, while it is part of a gesture that combines pitch with yaw for Wang and Ren. The use of a tabletop system required Forlines’ users to reach across the surface, resulting in particularly low pitch values [9].

Wang and Ren distinguished fingers, while other authors did not. They also recalibrated  $x/y$  offsets for every participant, thereby effectively using a per user calibration (Feng Wang, personal communication, 10/03/09). Finally, there are differences in how users commit a selection, such as take-off [13] or a button press with the other hand [3].

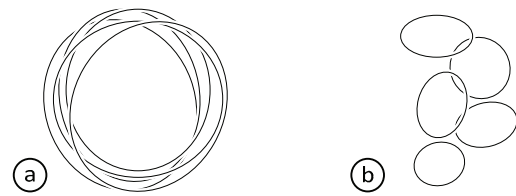
### Tracking Fingerprints

Fingerprint scanners have traditionally been used to identify users. More recently, they have also been used as parts of interactive systems. Sugiura and Koseki’s, for example, used a fingerprint scanner to allow users to invoke different functions by touching with different fingers [17].

Several patents explain how to control a pointer using a fingerprint scanner. Ferrari and Tartagni’s device allows controlling a mouse pointer as a relative touchpad [7]. Gust analyzes optical flow in order to extract motion [10]. Aki-zuki’s device implements an absolute touchpad [1]. A device by Bjorn and Belongie can distinguish whether it is being touched by a fingertip or a flat finger [6]. Unlike all of these patents, the *RidgePad* prototype presented in this paper is able to extract rolling and dragging from a single interaction (see also Figure 9), a crucial feature for the method proposed in this paper.

### USER STUDY 1: THE IMPACT OF USER ID AND FINGER POSTURE ON ACCURACY OF TRADITIONAL TOUCH

The generalized perceived input point model makes the assumption that the offset between the contact point and



**Figure 2: Main hypothesis. Expected outcome if touch inaccuracy is caused primarily (a) by the fat finger problem or (b) by the generalized perceived input point model.**

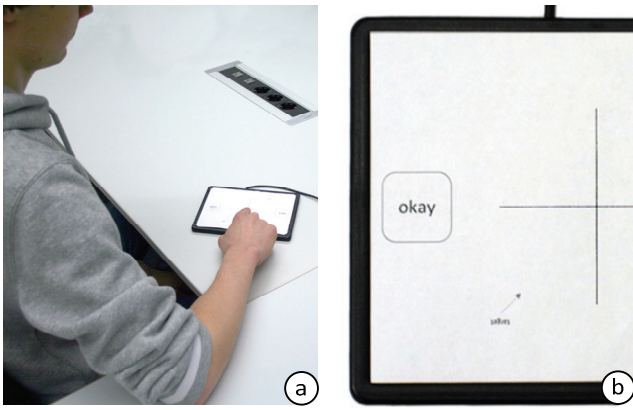
target depends on the *wider context* of the touch interaction, in particular roll, pitch, yaw, and user ID. The purpose of this study was to verify this assumption.

Our main hypothesis was that a variation of touch context, i.e., a variation of finger posture and/or user ID, would result in distinct clusters of touch positions. Figure 2 illustrates this. A participant has repeatedly acquired a target using five different finger postures. Each one results in a distribution, which we illustrate using an oval. If touch inaccuracy is governed primarily by the fat finger problem, we expect to see large ovals, all of which are centered on roughly the same point (Figure 2a). If the inaccuracy of touch, however, is primarily explained by the generalized perceived input point model, we expect to see ovals that are visibly *offset* with respect to each other (Figure 2b).

### Task

Figure 3 shows a participant during the study. A touchpad showed a single target, which participants acquired repeatedly. (There was no reason to include distracter targets. Distracters have a major effect on *adaptive* input techniques, such as magnetic targets (e.g., [4]), but not on unmodified touch). During each trial, participants first touched the start button on the pad (labeled “okay” in Figure 3). Then participants assumed the finger angle for the current condition with their right index finger and acquired the target. Participants committed the touch interaction by pressing a footswitch. This recorded the touch location reported by the touch pad, played a sound, and completed the trial. Participants did not receive any feedback about the location registered by the touchpad.

We took the following four measures to minimize the impact of other potential factors. First, participants kept their head in a fixed position above the touchpad, as shown in Figure 3. This controlled for parallax. Second, the crosshairs marking the target extended beyond participants’ fingers, allowing participants to maintain a certain amount of visual control during targeting. Third, the use of a footswitch allowed us to avoid artifacts common with other commit methods, such as inadvertent motion during take-off. And finally, participants were told to use as much time as necessary and that task time would not be recorded. Every participant completed their 600 trials in under 40 minutes.



**Figure 3: (a) A participant operating the touchpad. (b) The crosshairs mark the target.**

In the case of an accidental commit, such as activating the footswitch twice, the system discarded the input, notified participants with an acoustic signal, and had them repeat the trial.

### Finger angles: roll, pitch, and yaw

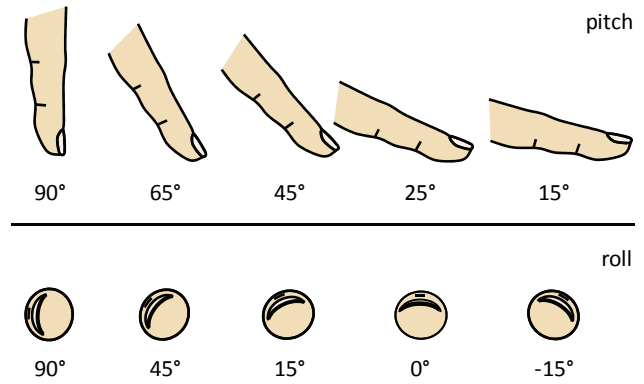
Participants acquired the target using their right index finger with five different levels of pitch and five different levels of roll (Figure 4). We varied pitch between “close to horizontal” =  $15^\circ$  and “straight down” =  $90^\circ$ . Pitch values beyond that caused the fingernail to touch first, which clashes with many types of capacitive devices. A roll of  $0^\circ$  meant that the nail was horizontal. We varied roll between “rolled slightly left” =  $-15^\circ$  and “rolled fully to the right” =  $90^\circ$ .

Varying roll and pitch separately allowed us to keep the number of trials manageable. During the pitch session, participants kept finger roll horizontal ( $0^\circ$ ), while they used a fixed pitch angle of  $15^\circ$  during the roll session. Combinations of pitch and roll are to be interpreted pitch-first. A pitch/roll of  $15^\circ/45^\circ$  thus means “assume a pitch of  $15^\circ$  and then roll the finger  $45^\circ$  to the right”.

We also studied the third angle, i.e., yaw. However, there was no need to vary it, because yaw takes places in the plane of the touchpad. As a result, we can reconstruct all levels of yaw by rotating the touch locations (obtained from a single level of yaw) post-hoc in software around the target. This, however, requires knowledge of the target location. Since the capacitive pad cannot see the target, we approximated its location by testing two levels of touchpad orientation ( $0^\circ$  and  $180^\circ$ ). We then determined the rotation center as the center of gravity among all touch locations, before we flipped the  $180^\circ$  condition and merged its samples with the  $0^\circ$  condition. To make sure that the  $180^\circ$  condition be identical from the participants’ perspective, participants operated a second ‘okay’ button on the opposite of the touchpad (cropped in Figure 3b).

### Procedure

To keep fatigue at a reasonable level, participants performed their trials in two sessions.



**Figure 4: Participants acquired targets holding their fingers in these finger pitch and finger roll angles.**

In one session, participants performed five variations of pitch  $\{15^\circ, 25^\circ, 45^\circ, 65^\circ, \text{ and } 90^\circ\}$ . In the other session, participants performed five variations of roll  $\{-15^\circ, 0^\circ, 15^\circ, 45^\circ, \text{ and } 90^\circ\}$ . Session order was counterbalanced across participants.

Within a session, participants performed a sequence of 150 trials with the touchpad in one orientation and then a second sequence of 150 trials with the touch pad in the opposite orientation. Pad rotation was counterbalanced across participants. For each sequence, participants completed 5 blocks of 5 angles  $\times$  6 repetitions each. The order of finger angles was counterbalanced across trial blocks.

Each participant completed all conditions. Each participant completed 5 angles  $\times$  2 pad orientations  $\times$  2 sessions  $\times$  5 blocks  $\times$  6 trials per block = 600 trials.

### Apparatus

The touchpad was a  $6.5'' \times 4.9''$  capacitive *FingerWorks iGesture* multi-touchpad. It was connected to an Asus eeePC 900HD. The footswitch was a Boss FS-5U.

### Participants

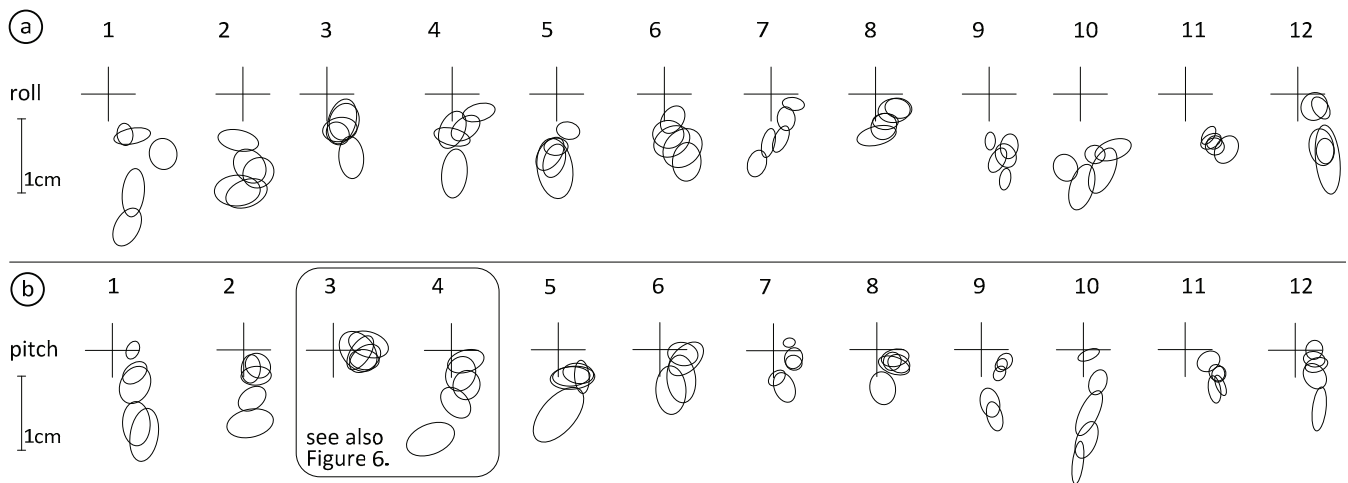
We recruited 12 participants (3 female) from our institution. All participants were right-handed and between 17 and 34 years old. Each received a small compensation as a gratuity for their time and we awarded €20 to the most accurate participant.

### Hypotheses

We had one main and four dependent hypotheses. Our *main hypothesis* was that a variation of touch context, i.e., a variation of finger posture and/or user ID, would result in significantly different clusters of touch positions. The dependent hypotheses spell this out for the individual variables.

1. **Pitch:** Different levels of pitch result in *distinct* touch location clusters. In other words, we expected to find higher spread across pitch levels than within a given pitch level.
2. **Roll:** analog to pitch.
3. **Yaw:** analog to pitch.



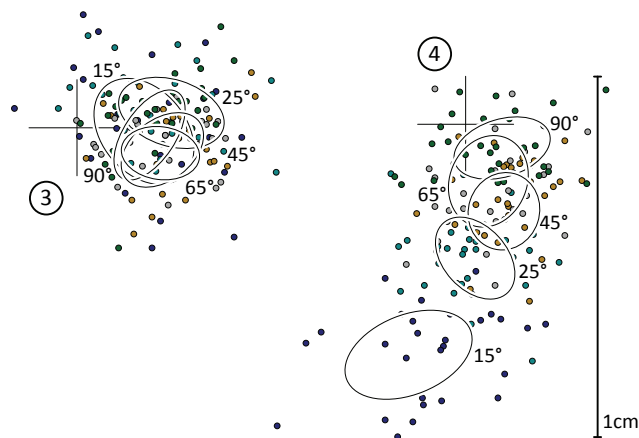


**Figure 5: Clusters of touch locations for each of the 12 participants (columns 1-12). Crosshairs represent target locations; ovals represent confident ellipsoids. (a) Each of the 5 ovals represents one level of roll. (b) Each of the 5 ovals represents one level of pitch. All diagrams are to scale. Note how different patterns suggest that each participant had a different interpretation of touch.**

- User:** Cluster organization will differ across participants. Different users have different finger shapes and we hypothesized they might also have different mental models of how to map their large fingers to a small target.

### Results

Figure 5 summarizes the complete touch location data obtained from this study. Each column summarizes the recorded locations for one participant; the top chart shows aggregated clusters of touch locations for the different levels of roll, the bottom chart shows the aggregation of the pitch session. All ovals in Figure 5 represent confidence ellipsoids that contain 65% of the recognized touch locations per condition. The crosshairs in each chart is the target location. Figure 6 shows two examples in additional detail (pitch data of Participants 3 and 4).



**Figure 6: Close-up of touch locations organized by pitch of Participants 3 and 4 from Figure 5b. Even though clusters are much further apart for Participant 4, both are equally “accurate” under the generalized perceived input point model.**

### Pitch

We analyzed the effect of pitch using a repeated measures one-way ANOVA. To better understand the nature of the differences, we decomposed the differences in recognized touch position into differences *along* the finger axis (y axis in the chart) and *across* the finger axis (x axis in the chart). Changing finger pitch had a significant effect on recognized touch positions ( $F_{4,8}=6.620, p=0.012$ ) *along* the finger axis. Pair-wise comparisons using Bonferroni-corrected confidence intervals showed that the touch locations of all levels of pitch were significantly different (all  $p<0.05$ ). We also found a main effect of pitch on touch location *across* the finger axis ( $F_{4,8}=6.972, p=0.01$ ). However, pair-wise post-hoc tests showed no significant differences.

### Roll

A repeated measures one-way ANOVA found a significant main effect of roll on sensed touch position *along* the finger axis ( $F_{4,8}=4.574, p=0.032$ ). Bonferroni corrected pair-wise comparisons showed a significant difference between 90° roll and all other roll levels, as well as 45° vs. -15° and 0° (all  $p<0.05$ ). An ANOVA on touch location *across* the finger axis did not find an effect ( $F_{4,8}=1.444, p=0.305$ ).

### Yaw

We ran paired-samples *t*-tests comparing touch locations *across* and *along* the finger axis in the two yaw conditions. We found both to be significantly different (*across*:  $t_{11}=6.570, p<0.001$ ; *along*:  $t_{11}=9.361, p<0.001$ ).

### Participant

We ran a two-way ANOVA on finger pitch and participant both *along* and *across* the finger axis, using participant as a random factor. We found a significant interaction between pitch and participant and significant main effects for both (all  $p<0.001$ ). For each participant, we ran separate one-way ANOVAs on finger pitch to determine where the effect was particularly evident. We found a significant main effect on touch location *along* the finger axis for all participants

and a significant main effect *across* the finger access for all but one participant (all  $p < 0.05$ ).

Similarly, we ran a two-way ANOVA on finger roll and participant. We found significant main effects for participant along and across the finger axis, as well as for finger roll along the finger axis. We further found a significant interaction between finger roll and participant along and across the finger axis (all  $p < 0.001$ ). This indicates that each participant exhibits a different behavior and touch pattern in response to finger roll. We ran one-way ANOVAs on finger roll separately for each participant. We found a significant main effect of finger roll both along and across the finger axis for all participants (all  $p < 0.05$ ), except one whose error rates did not differ significantly across the finger axis.

## Discussion

### Finger angles

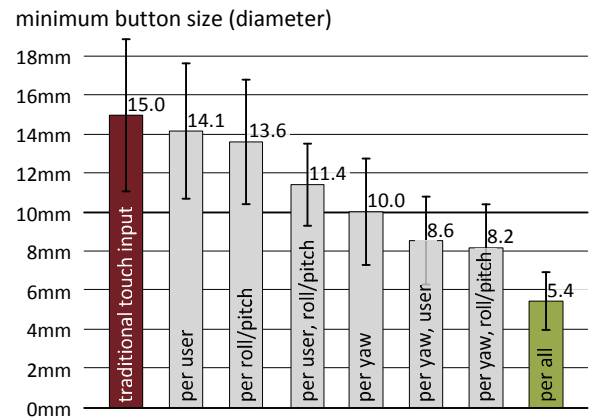
As hypothesized, all three angles had an impact on touch location and lead to distinct clusters, supporting hypotheses 1-3. As expected based on Forlines et al. [9], finger pitch primarily impacted touch location along the finger axis (visible as vertical patterns in Figure 5, bottom row). A “flatter finger” caused the touchpad to locate targets farther away from the target towards the user’s palm. Somewhat surprisingly, variations in *roll* impacted touch location primarily *along* the finger axis as well, more than across (visible as vertical patterns in Figure 5, top row). Finally, as expected, there also was a significant effect of yaw on touch location. This is also obvious in Figure 5 where none of the groups of ovals are centered on the target. This emphasizes the fact that global offsets, as applied by Vogel and Baudisch [18] need to consider hand yaw.

### Users

Also as hypothesized, there was an effect of user on the touch location. As shown in Figure 5, the clusters of recognized touch positions varied across participants, and they did so quite substantially. Figure 6 shows a particularly different pair: For Participant 4, touch locations vary drastically with pitch, while pitch has very little impact on the touch locations produced by Participant 3.

Based on this chart, one might think that Participant 3 is simply more *accurate* than Participant 4, e.g., that Participant 3 performed the task with additional care. Whether this is true or not is a matter of perspective. When we look at the size of the individual clusters of the two users, we see that they are roughly comparable. This means that both participants reproduced the target location equally well. What differs between the two participants is their mental model. Participant 3’s understanding of touch coincides strongly with the capacitive touchpad model.

So while Participant 3 is more fit than Participant 4 when operating *today’s* touch devices, when using an input device based on the generalized perceived input point model this is not the case anymore. As we explain in the following sections, such a device compensates for roll, pitch, yaw, and user ID. “Accuracy” now means neither the proximity



**Figure 7: Minimum size of a button that contains 95% of all touches on a touch device that knows about different subsets of roll/pitch, yaw, and user ID. Error bars encode standard deviation across all samples.**

of a cluster to the target (because we can compensate for it), nor the proximity of clusters to each other (again, because we can compensate for it). Instead, accuracy now means size of clusters, as all other factors can be compensated for. Since the cluster sizes for Participants 3 and 4 are comparable, this means both participants will perform equally well under the new model.

### Main hypothesis

Overall, and most importantly, our study supports our main hypothesis: roll, pitch, yaw, and user ID all lead to *distinct* clusters (i.e., significantly different centroids). As a matter of fact, these clusters are clearly separated, as discussed earlier when explaining Figure 2. Our findings therefore support that the generalized perceived input point model indeed explains a significant part of the inaccuracy of touch, rather than the fat finger problem.

### Exploiting the model with a device

These observations suggest that a device should be able to obtain improved accuracy by applying compensating offsets for each condition.

The data from our study allows us to make predictions about the performance such a device might achieve. Figure 7 shows a summary. Each bar denotes the diameter of a round button that contains 95% of all touches, assuming that we apply compensating offsets for different subsets of factors. Each bar was computed by mapping the centroids of different sizes of clusters to the target center location. For the traditional touchpad condition (left bar), no mapping was applied. For the “per yaw” condition, the centroid of all touches was moved to the target. For the ‘per yaw and roll/pitch’ condition, the centroids of each roll cluster and each pitch cluster were moved to the target. For the ‘per all’ condition, the centroids of each roll cluster and each pitch cluster for each participant were moved to the target.

As illustrated by the chart, each additional piece of information should allow the device to further improve its accuracy up to a factor of 2.75 if all angles and user ID are included.



**Figure 8:** The *RidgePad* prototype is based on an *L SCAN Guardian* fingerprint scanner.

Instead of buttons measuring 15mm, such a device should allow users to reliably acquire buttons measuring 5.4mm.

A factor of 2.75 suggests considerable potential. In order to exploit it, however, we need a touch device capable of extracting these four additional variables from a touch interaction. We have created two prototypes of such devices. One of them is *RidgePad*.

#### **RIDGEPAD: A HIGH-PRECISION TOUCH DEVICE**

*RidgePad* is a touch input device that implements the requirements of the generalized perceived input point model based on a fingerprint scanner.

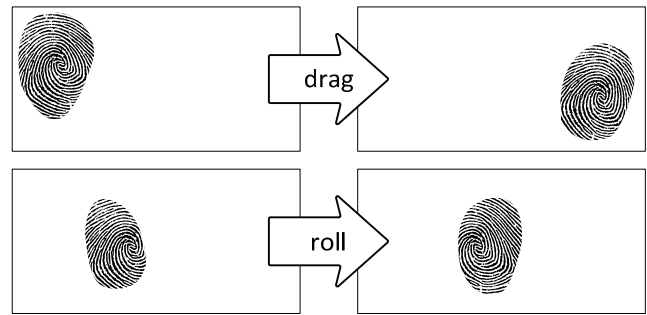
Traditional touch devices, such as the FingerWorks pad obtain only the contact area of the finger with the surface. *RidgePad* obtains the same contact area—from a fingerprint scanner, however (a high-resolution optical *L SCAN Guardian*, see Figure 8). The fingerprint offers two additional types of information. First, it allows *RidgePad* to identify the user. Second, it allows *RidgePad* to analyze the portion of the user's fingerprint that is located inside the contact area. Based on its analysis of which part of the fingerprint touches the screen (Figure 1b), *RidgePad* infers all three finger angles, i.e., yaw, pitch, and roll.

This mechanism allows *RidgePad* to extract rolling and dragging from a single interaction, as shown in Figure 9. (a) When dragging, fingerprint outline and features move in synchrony (i.e., features do not change their position relative inside the fingerprint). (b) When rolling the finger on the surface, fingerprint features remain stationary.

#### **Algorithm: calibration**

By default, *RidgePad* is only as good as a regular touch pad. It achieves improved accuracy through calibration.

During calibration, users repeatedly acquire a single target on the fingerprint scanner. It is not necessary for users to touch under specific roll, pitch, and device rotations; the more postures users cover, however, the more postures will benefit from improved precision.



**Figure 9:** (a) When dragging, fingerprint outline and features move in synchrony. (b) When rolling the finger on the surface, fingerprint features remain stationary.

Every calibration trial produces a pair of a fingerprint image and an associated target position relative to the center of the fingerprint, i.e., an offset. All pairs are stored in the user's profile.

The profile is user-specific, but not device-specific. This allows users to calibrate future devices instantly using an existing profile that is associated with their fingerprint.

#### **Algorithm: usage as a touch device**

During actual use, users touch *RidgePad*'s surface just like any other touch device. *RidgePad* computes the center of the contact area as a reference point. It then compares the observed fingerprint with all fingerprints in its database (the search space is reduced to the user's profile as soon as the user has been identified). *RidgePad* compares fingerprints using the generic image-matching algorithm SURF by Bay et al. [5]. SURF extracts features from images, such as intersections of lines. It then finds the image transformation that maximizes the number of features that line up.

The number of fingerprints in the profile that have *some* match with an observed fingerprint is typically large. To determine which fingerprints are most likely to represent the pitch and roll position of the observed fingerprint, *RidgePad* simply uses the number of features SURF was able to match as a metric. This works because two images are likely to exhibit similar features if and only if similar parts of the finger touched the surface. *All* features typically match only if pitch, roll, and yaw were identical.

Based on this similarity function, *RidgePad* looks up the  $k$  closest matches in the user's profile ( $k$ -nearest neighbor algorithm). *RidgePad* then averages the offset values associated with the chosen neighbors (optionally with additional weight for better matches) and finally adds that offset to the center of the current touch location.

#### **Hardware Implementation**

The *Guardian* fingerprint scanner in our prototype offers a 3.2" × 3.0" touch area. As common for fingerprint scanners, it works based on frustrated total internal reflection (FTIR). Unlike FTIR implementations in current tabletop or wall systems such as [12], the glass surface is illuminated from *below* and its illumination is frustrated by the finger. The



Guardian provides images at 500ppi, which allows for high-quality fingerprint recognition.

### Benefits and Limitations

The generic nature of its algorithm makes RidgePad particularly robust and flexible. RidgePad finds matches for a given yaw/pitch/roll/user fingerprint, because it finds other fingerprints that “look” similar; nowhere in the system are they ever labeled with angles. While we designed the algorithm to work with roll, pitch, and yaw, it is independent of any such specifics. It should therefore be straightforward to extend the algorithm to other features, such as finger pressure.

One of the limitations of the current implementation is time complexity. The *Guardian* fingerprint scanner in our prototype requires a noticeable pause before transmitting a picture. In addition, our non-optimized prototype code sequentially compares fingerprints with all fingerprints in the user’s database, which takes 200-300ms for each comparison. Future versions should be able to achieve real-time performance by extracting features up-front and comparing feature vectors using more suitable data structures.

### USER STUDY 2: TOUCH PRECISION WITH RIDGEPAD

To verify the performance of touch devices based on the generalized perceived input point model, we conducted a second user study. We compared RidgePad and a device design based on an optical tracker with a traditional baseline condition. Similar to the first study, we analyzed the effect of roll, pitch, and yaw. Our main hypothesis was that RidgePad and the optical tracker would outperform the baseline condition.

### Interfaces

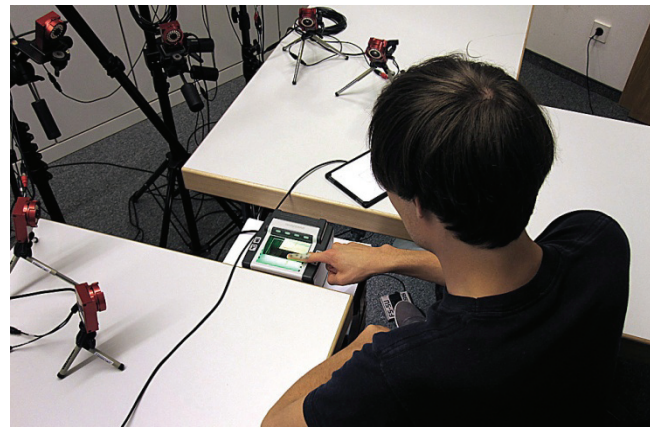
We tested three interface conditions, all of which were implemented by the hardware setup shown in Figure 10. This particular setup allowed us process every targeting trial with each of the three interfaces simultaneously.

*The fingerprint interface* was implemented using RidgePad and employed the algorithm described in the previous section.

*The control interface* simulated a traditional touchpad interface. It received the same input from the fingerprint scanner as the fingerprint interface. However, this condition did not use the fingerprint features and instead reduced the fingerprint to a contact point at the center of the contact area.

*The optical tracker interface* was implemented based on a six-degree of freedom optical tracking system (an 8-camera *OptiTrack V100* system). To allow the system to track the participant’s fingertip, we attached five 3-mm retro-reflective markers to the participant’s fingernail (Figure 10). The extreme accuracy of the optical tracker made this interface a “gold standard” condition that allowed us to obtain an upper bound for the performance enabled by the generalized perceived input point model.

Similar to the fingerprint interface, the optical tracking interfaces applied corrections by averaging offsets from



**Figure 10: The three interfaces: the fingerprint scanner simultaneously implemented the *fingerprint interface* and the *control interface*. The red cameras below the *optical tracker interface*, which was based on *OptiTrack VT100* cameras. Between trials, participants tapped the touch pad. They committed using the footswitch.**

$k=13$  training samples that matched in terms of roll, pitch and yaw. The optical tracker interface, in contrast, obtained these angles from direct optical measurement of the markers position in 3-space.

Since *fingerprint interface* and *optical tracker interface* required per-user calibration, we used 80% of all trials (520 of 650) as training data for the respective calibration procedures. We used the remaining 20% of all trials (130) for the actual analysis.

### Task

As in our first study, participants acquired a single target repeatedly. The target was drawn onto the surface of the fingerprint scanner. Half of all participants acquired a target marked with crosshairs similar to our first study. The other half of participants acquired a target marked with only a dot. The additional independent variable crosshairs vs. dot allowed us to study the impact of the occlusion problem. As in the first study, participants pressed “okay”, acquired the target, and committed using a footswitch. All participants completed all trials of one session in about 30 minutes.

### Procedure

Participants completed the same roll/pitch combination as in the first user study plus four additional variations of roll across  $45^\circ$  of pitch (Figure 11, additions in bold).

Participants completed the study in two sessions; the second session was identical to the first, except that we rotated the scanner for the same reasons as in our first study. We counterbalanced the order of all conditions within sessions as well as sessions and rotations across participants. Overall, participants completed 2 sessions  $\times$  5 blocks  $\times$  5 repetitions  $\times$  13 angles = 650 trials.

### Participants

We recruited a fresh set of 12 participants (2 female) from our institution. All participants were right-handed and between 22 and 34 years old. Again, we gave each participant



roll	-15°	0°	15°	45°	90°
pitch					
15°	×	×	×	×	×
25°		×			
45°	×	×	×	×	×
65°		×			
90°		×			

**Figure 11: Study conditions. Angles for pitch and roll from which participants had to acquire the target.**

a small compensation for their time and awarded €20 to the most precise participant.

### Apparatus

The fingerprint scanner and the optical tracker were powered using an Intel Core 2 Duo machine running at 3GHz with 3GB of RAM. We reused both the footswitch and the FingerWorks pad from our first study; however, the latter was used for the “okay” button only this time.

### Hypotheses

We had two hypotheses:

1. Optical trackers track angles with extremely high precision. We therefore expected the *optical tracker interface* to redeem the entire accuracy benefit suggested by the first study, i.e., an improvement of a factor of 2.75 compared to the simulated capacitive *control interface*.
2. RidgePad cannot reconstruct angles quite as accurately as an optical tracker. Still we expected the *fingerprint interface* to improve input precision substantially compared to the simulated capacitive control condition.

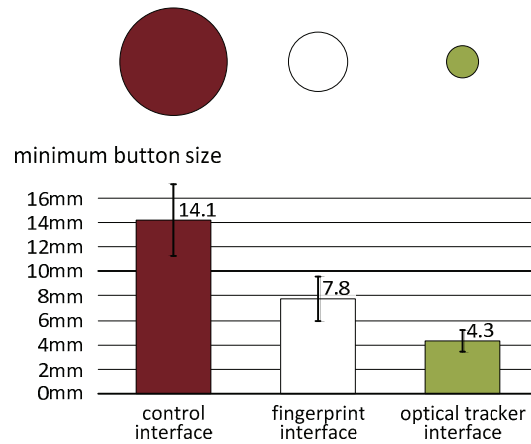
We were curious to see how large the improvement of the *fingerprint interface* would be compared to the *control interface*.

### Results

Similar to the analysis of our first study, we compared the spread of recorded input locations (i.e., the mean distance of all points in a population from their center of gravity). We compared the mean input spread for each participant when using each interface.

We ran a one-way ANOVA on averaged per-participant spread with participant as a random variable and found a significant main effect of interface on spread ( $F_{2,9}=49.457$ ,  $p<0.001$ ). Pair-wise comparisons using Bonferroni-corrected confidence intervals showed statistically significant differences of spread between all interfaces ( $p<0.01$ ). The *control interface* caused the largest amount of average spread (2.75mm), followed by fingerprint-corrected locations (1.24mm). Locations corrected with the *optical tracker interface* had the lowest average spread (0.85mm). This means that the spread of touch input after fingerprint-based correction was on average 2.2 times smaller than when uncorrected. On average, optical-tracker-based corrections brought down spread by a factor of 3.3 compared to the *control interface*.

to-scale minimum button



**Figure 12: Minimal target sizes to achieve a 95% success rate. The circles are to-scale representations of the respective minimum target sizes. Error bars encode standard deviations.**

### Dot Targets vs. Crosshairs Targets

Average spread for each interface was 1.9mm/1.4mm/0.9mm for the crosshairs conditions and 2.2mm/1.9mm/1.2mm for the dot conditions. A two-way ANOVA, however, did not find a significant interaction between target type and interface ( $F_{2,9}=1.44$ ,  $p=0.287$ ). We did not find a main effect of target type on spread either ( $F_{1,10}=3.186$ ,  $p=0.105$ ). The fact that *dot* targets performed successfully as well, however, suggest that the methods proposed in this paper also apply to targets that are subject to the *occlusion problem* [18].

### Discussion

This study supports our claim that touch devices can increase accuracy by exploiting the *generalized perceived input point model*.

Figure 12 shows another perspective on the results. It shows the minimum target sizes that users can acquire with 95% accuracy for each of the three interfaces. Sizes were computed so as to include 95% of all touches across participants and conditions (spread across all participants, sessions, and roll/pitch conditions plus 2 standard deviations).

The circles on top of Figure 12 illustrate the resulting buttons *to-scale*. The *fingerprint interface* achieves a minimum target size 1.8 times smaller than the *control interface*. The *optical tracker interface* reduces target size by a factor of 3.3. The resulting button occupies less than 10% of the size of the *control interface* button.

### CONCLUSIONS AND FUTURE WORK

In this paper, we made two types of contribution.

On the one hand, we made a technical contribution. The *RidgePad* device achieved 1.8 times higher accuracy than simulated capacitive and we demonstrated that the use of high precision 3D tracking can more than triple touch accuracy. This substantial increase in accuracy can be used to make touch interfaces more reliable or to pack up to 10 times more controls into the same touch surface. Future

versions of the optical tracking device might achieve a smaller footprint by using cameras placed in the corners of the screen. A more distant future version of RidgePad might leverage *in-cell* technology to bring touch input to very small mobile devices, such as interactive watches and rings.

On the other hand, we made a contribution in the theoretical domain, which we tend to think of as at least equally important. We introduced a new model for touch inaccuracy, the *generalized perceived input point model*. We presented a user study, the findings of which are congruent with our new model, while they refute the *fat finger problem*, which was traditionally considered the primary source of touch inaccuracy.

This paper also contributes a new perspective on touch. Touch has traditionally been considered a 2D phenomenon, most likely because touch screen interaction required only two coordinates, i.e., an *x/y* coordinate pair. The proposed model, in contrast, establishes touch as a phenomenon of not only the touch surface, but of a wider context of three-dimensional factors. While we primarily investigated the user's finger posture in 3-space, this wider context may include additional factors, such as head position, device orientation, parallax, and so on. Tracking these additional factors might allow future devices to realize even larger improvements in touch accuracy. Additional research is required here.

Finally, we learned about users. We found that users are not inaccurate—they are just different. The most likely explanation for this difference is that touch *on a millimeter scale* was never defined in the first place. For targets on this almost microscopic scale, pointing means to “dock” a comparably large, asymmetric object with a tilted surface. Comparing the arrangement of ovals across Figure 5 clearly shows that no two participants of our study had the same mental model of how to accomplish this. Contact area—the determining factor in today's touch technology—might be a factor, but clearly only one of many. Additional research is required to reach a new and more detailed understanding of users' mental models of touch.

#### ACKNOWLEDGMENTS

We thank *CrossMatch* for providing us with a *Guardian* Fingerprint Scanner and silicone sheet. We also thank Sean Gustafson for valuable comments and constant support. We thank all participants of our studies and numerous pilots for their time. Thanks also to Chris Harrison for his comments.

#### REFERENCES

1. Akizuki, S. Touch pad having fingerprint detecting function and information processing apparatus employing the same. US Patent 6,360,004. 05/19/2002.
2. Albinsson, P. Zhai, S. 2003. High precision touch screen interaction. In *Proc. CHI'03*, 105–112.
3. Baudisch, P. and Chu, G. 2009. Back-of-device interaction allows creating very small touch devices. In *Proc. CHI'09*, 1923–1932.
4. Beaudouin-Lafon, M. and Mackay, W. E. 2000. Reification, polymorphism and reuse: three principles for designing visual interfaces. In *Proc. AVI'00*, 102–109.
5. Bay, H., Tuytelaars, T., Van Gool, L. 2006. SURF: Speeded Up Robust Features. In *Proc. ECCV'01*, 404–417.
6. Bjorn, V.C. and Belongie, S.J. Configurable multi-function touchpad device. US Patent 6,950,539. 09/27/2005.
7. Ferrari, A. and Tartagni, M. Touchpad providing screen cursor/pointer movement control. US Patent 6,392,636. 05/21/2002.
8. FingerWorks SDK, <http://www.fingerworks.com>
9. Forlines, C., Wigdor, D., Shen, C., and Balakrishnan, R. 2007. Direct-touch vs. mouse input for tabletop displays. In *Proc. CHI'07*, 647–656.
10. Gust, L.A. Compact optical pointing apparatus and method. US Patent 7,102,617. 09/05/2006.
11. Hall, A.D., Cunningham, J.B., Roache, R.P., and Cox, J.W. 1988. Factors affecting performance using touch-entry systems: Tactual recognition fields and system accuracy. In *Journal of Applied Psychology*, 4, 711–720.
12. Han, J. Y. 2005. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proc. UIST'05*, 115–118.
13. Potter, R., Weldon, L., Shneiderman, B. 1988. Improving the accuracy of touch screens: an experimental evaluation of three strategies. In *Proc. CHI'88*, 27–32.
14. Ren, X., Moriya, S. 2000. Improving selection performance on pen-based systems: a study of pen-based interaction for selection tasks. In *TOCHI*, 7(3):384–416.
15. Roudaut, A., Huot, S., and Lecolinet, E. 2008. TapTap and MagStick: Improving One-Handed Target Acquisition on Small Touch-screens. In *Proc. AVI'08*, 146–153.
16. Roudaut, A., Lecolinet, E., and Guiard, Y. 2009. MicroRolls: expanding touch-screen input vocabulary by distinguishing rolls vs. slides of the thumb. In *Proc. CHI'09*, 927–936.
17. Sugiura, A., Koseki, Y. 1998. A user interface using fingerprint recognition: holding commands and data objects on fingers. In *Proc. UIST'98*, pp 71–79.
18. Vogel, D. and Baudisch, P. 2007. Shift: A Technique for Operating Pen-Based Interfaces Using Touch. In *Proc. CHI'07*, 657–666.
19. Wang, F., Cao, X., Ren, X., and Irani, P. 2009. Detecting and leveraging finger orientation for interaction with direct-touch surfaces. In *Proc. UIST'09*. 23–32.
20. Wang, F. and Ren, X. 2009. Empirical evaluation for finger input properties in multi-touch interaction. In *Proc. CHI'09*. 1063–1072.