

Measuring the Linear and Rotational User Precision in Touch Pointing

Francois Berard

University of Grenoble
LIG, Grenoble-INP
BP 53, 38041 Grenoble Cedex 9, FRANCE
francois.berard@imag.fr
+33-476-514-365

Amelie Rochet-Capellan

GIPSA-lab, UMR 5216 CNRS
Grenoble INP/UJF/U. Stendhal
BP 46, 38402 Grenoble Cedex, FRANCE
ameliecapellan@free.fr
+33-476-574-850

ABSTRACT

This paper addresses the limit of user precision in pointing to a target when the finger is already in contact with a touch surface. User precision was measured for linear and rotational pointing. We developed a novel experimental protocol that improves the estimation of user's precision as compare to previous protocols. Our protocol depends on high-resolution measurements of finger motions. This was achieved by the means of two optical finger trackers specially developed for this study. The trackers provide stable and precise measurements of finger translations and rotations. We used them in two user experiments that revealed that (a) user's precision for linear pointing is about $150dpi$ or $0.17mm$, and (b) user can reliably point at sectors as narrow as 2.76° in $2s$ in rotational pointing. Our results provide new information for the optimization of interactions and sensing devices that involve finger pointing on a surface.

Author Keywords

Touch, Precision, Human Precision, Finger Rotation, Computer Vision, Tracking

ACM Classification Keywords

H.1.2 [User/Machine Systems]: Human factors; H.5.2 [User Interfaces]: Input devices and strategies

General Terms

Measurement, Performance, Experimentation, Human Factors

INTRODUCTION

With the emergence of large interactive surfaces, and the widespread adoption of mobile interactive devices (e.g., smartphones, tablets), touch interaction is becoming a major way to interact with computers. As touch devices evolve into fully capable computers, with display's pixel count matching

that of desktop monitors¹, they can support complex tasks, which require increased input precision from users. In this context, a challenging question to address is users' precision in finger pointing gestures. In this paper, we report new methods and empirical results to evaluate the precision achievable by users with touch interaction, both in linear and rotational pointing.

Direct touch pointing is the prime interaction for touch surface. In direct touch, the finger is initially *above* the touch surface. The target is acquired upon landing the finger directly on top of it (or shortly after, on takeoff). This mode of pointing is characterized by a *discrepancy in the location of the active point* of the finger, as imagined by the user and as measured by the touch surface. Holz et al. studied this phenomenon carefully and showed that, with an ideal sensing mechanism and user calibration, targets have to be more than $4mm$ in diameter for reliable selection [11, 12]. This is 40 times the size of pixels of recent touch computer displays¹. In other words, direct touch is not immediately usable for precise tasks such as positioning a cursor between letters, or acquiring small targets in a dense display. One way to allow accurate pointing with touch devices is to use a virtual pointer controlled by finger motions when the *finger is already in contact* with the device [1, 3, 21, 24]. With the user focusing on the position of the virtual pointer, instead of some particular point of the finger, the active point discrepancy issue disappears. The achievable precision is then restricted by other factors such as the sensing device precision, the visual acuity, the interaction technique involved, and the limit of human's precision when moving a finger in contact with a physical surface. Our work focuses on the latter.

The first contribution of this work is to provide a new method to estimate the smallest targets *in the motor space* that can be acquired by a finger already in contact with a surface. We focus on user's motor capabilities. We neutralize the effect of the visual acuity and interaction technique factors by providing a magnified graphical feedback (Figure 6), and by using the identity as the transfer function between motor and visual spaces (constant gain = 1). In order to neutralize the device sensing precision factor, we developed the second contribution of this work: two high precision trackers, one for linear finger motions and one for finger rotations. The precisions of these trackers are substantially better than users' precision.

¹Recent tablet computers carry displays with 2048x1536 pixels, see for example <http://www.apple.com/iPad/>

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, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ITS'12, November 11–14, 2012, Cambridge, Massachusetts, USA.
Copyright 2012 ACM 978-1-4503-1209-7/12/11...\$15.00.

The third contribution of this work is to provide first estimations of user’s precision in linear and a rotational pointing.

Users’ maximum achievable precision is key information for *interaction designers* of touch systems. It is informative to design pointing interactions on very small devices [2], or indirect input technique for games on mobile devices (e.g. virtual joystick, slingshot, etc.). It provides an estimate of the “limb precision” required to optimize the C/D gain[6] or to design subpixel interactions [22]. It also defines the requirement for the precision of any touch sensing device and, as such, it is an important information for *input device designers* as well: if we can state that a device already saturates users’ precision, research effort should be allocated on other parameters than the precision of the device. Conversely, a device that measures finger pointing with less precision than users’ abilities is introducing a bottleneck in the input bandwidth. This is especially important in the case of the finger orientation, which has many potential applications in novel interaction designs [17, 20, 25], but for which the empirical knowledge on user’s precision is limited.

After a review of the previous work, we present a new experimental protocol meant to estimate user’s precision. We then introduce the two optical finger tracking techniques aimed at very high precision and low latency measurement of finger linear and rotational motions, as required by the experimental protocol. These trackers and protocol were used in two user studies that provide first empirical estimations of the limits of user’s precision in linear and rotational touch pointing. Finally, we discuss the results of the experiments and conclude.

PREVIOUS WORK

The limit of user’s precision in manipulating input devices has been investigated through various approaches. Chapuis et al. systematically questioned the effect of visual scale, motor scale and quantization of the acquisition of small targets [7]. They found that users’ performance degraded at small scale, in particular in the motor space and for target size in the range $0.1 - 0.2mm$. Casiez et al. studied the impact of the control-display gain on user’s performance. They identified a “limb precision problem” when using a high CD gain for reciprocal target acquisition [6] and evaluated, the resolution of the hand and fingers at $0.2mm$. In Chapuis et al. and Casiez et al., the limit of precision in the motor space was not the main focus, and the input device was a high-resolution computer mouse. Rahman et al. developed a more systematic approach to estimate the limit of users’ precision in wrist rotations for wrist-based input [19]. In their protocol, targets were defined on various discretization of a semi-circle. Berard et al. compared users’ precision when pointing with a mouse, to user’s precision when pointing with two 3D devices [4]. They found wide variations in user’s precision according to the device. Berard et al.’s experimental protocol also involved a discretization of target sizes. In the next section, we point to the issue of the discretization approach and describe a new experimental protocol that solves this issue.

To our knowledge, the limit of user’s precision in finger pointing while in contact with a touch device has never been

specifically studied. Studying the limit of user’s finger pointing precision requires a precise device to track finger motion. In a recent work, Rogers et al. showed that both the yaw and the pitch angle of a finger landing on a touch surface could be inferred from the signal sent by a capacitive sensing device [20]. Yet, their system was designed in order to improve the precision of pointing from above the surface. Wang et al. also inferred finger orientations from the higher resolution signal provided by an FTIR touch surface [25]. They estimated the precision of their system to $\pm 1^\circ$, which is still too broad for our purpose. Inferring parameters from a rough signal is interesting when one is constrained by the input device. However, an accurate estimation of user’s precision requires a precise input device. Many finger tracking approaches have used direct visual sensing (see Erol et al., for a review of several approaches reported in the literature [8]). Most of these visual finger tracking systems, though, are directed towards the creation of novel forms of interactions, not towards measuring precise positions. For example, Oka et al., used a thermal camera to facilitate the segmentation of the hand from the background [18]. They analyzed the extracted information for finger pointing and multi-finger gestures recognition. Yet, thermal imaging is restricted in term of resolution and sampling rate: their camera provided only 512×512 pixels at $30Hz$. Letessier et al. detected the position of many fingers in a multi-users scene recorded with a regular visual camera. They used background subtraction and shape filtering [14], but they did not estimate the orientations of the fingers. Malik et al. measured fingertip position and finger orientation from two webcams positioned close to the interactive surface [17]. Their system also relied on background subtraction to extract the contour of the hands. The contour was then filtered to infer the position and the orientation of the fingers. In the present work, we build on this last approach to develop a high precision tracking system. The precision of the tracker was increased by using a more precise camera and by improving the models of the finger contour.

MEASURING THE LIMIT OF USER PRECISION

The central idea to measure user’s precision limit for a given task is to find a threshold in the task precision above which user’s performance drops, showing that user’s precision limit has been reached. Fitts’ law [9] is a powerful tool to evaluate user’s performance in pointing tasks. Fitts’ law predicts that the mean time to acquire a target (MT) is a function of a single parameter: the ratio between the initial distance to target, also called *amplitude* (A), to the target *width* (W). Fitts’ law also predicts a linear relationship between MT and the Index of Difficulty (ID) as expressed in the following equations [15]:

$$ID = \log_2(A/W + 1) \quad (1)$$

$$MT = a + b.ID \quad (2)$$

Assuming that there is a width below which targets become significantly more difficult to acquire than “normal” targets, mean acquisition time should become significantly higher than the Fitts’ prediction for these targets and Equation 2 should be broken. This break in linearity could be taken as an indicator of the saturation of users’ precision.

Limits of previous approaches

Following this rationale, Berard et al. designed an experimental protocol to evaluate user's precision for 3 different devices [4]. In their protocol, participants acquired targets of various sizes, some assumed to be above and some below the limit of user's precision. The authors expected to observe the linear relation of Equation 2 for target sizes above the threshold, but to witness acquisition times significantly above the prediction for sizes below the threshold. This was the case for the mouse, while the results for the two other devices were not as clear, and required to consider both MT and the error rate to evaluate the threshold.

We identified two shortcomings in this protocol. The first one is related to the discretization of target size: the user's precision limit is bracketed by two of the experiment's target widths. Because participant time is constrained, only a limited number of target widths can be included in the experiment, resulting in a broad estimation of the user's precision limit. In other words, the protocol does not provide a direct estimation of the user's precision limit. For example, if participants' performance drops between targets of width 2 and 3 ticks (the smallest reported motion of the device) for a 1200dpi device, then the estimation of the user's precision limit is in the rather broad range 400 – 600dpi.

The second shortcoming of Berard et al.'s protocol is related to its sensitivity to the speed-accuracy tradeoff. Careful participants will fight to acquire the targets that are above their precision ability, at the cost of an increase of the acquisition duration. In contrast, faster and less careful participants will simply race through difficult targets, yielding no significant time increase, but substantially increasing their error rate. As participants in an experiment tend to be distributed along this speed-accuracy scale, the identification of the performance drop is blurred between an increase in the acquisition time and the error rate. The speed-accuracy tradeoff is handled in classical Fitts' experiments using the *effective width* [15] (ew) achieved by the participants, instead of the requested target width (W), in the computation of ID (Equation 1). When the distribution of error (final distance to target) is known, the effective width is computed according to Equation 3.

$$ew = 4.133 \times stddev(error) \quad (3)$$

With the effective width, slow and careful participants will have ew smaller than W , while fast participants will have larger ew . This approach does not impose a speed-accuracy tradeoff to participants, but it takes into account the tradeoff adopted by each participant into the computation of ID . This is fine for regular Fitts' experiments, where the goal is to improve the fit of Equation 2 to experimental data. When searching for user's precision limit, what is needed is a homogenization of the participants' strategies.

A new protocol: unpredictable controlled durations

In order to solve the problems of speed-accuracy tradeoff we impose the duration of the pointing. Imposing a pointing duration puts every participants on the same level for a given task: they all have to make the best of the time they are

given to point to the target. In particular, participants cannot slow down the task for accuracy, nor speed it up for efficiency. Hence the control of the speed-accuracy tradeoff is transferred from the participant to the experimenter. Indirectly, controlling time also solves the discretization problem by switching the roles between pointing time and target width in the experimental design. Pointing time becomes a factor controlled by the experimenter, and target width becomes a dependent variable measured from the participant's performances.

In our protocol, participants point at the smallest possible target (1 tick width) with as much accuracy as possible within the allocated duration, i.e. we ask them to minimize the distance between the pointer and the target. This distance is recorded at the end of the duration controlled by the system, when the target disappears. As a consequence, there is no validation from the participants at the end of a trial and so, no *error rate*. Thus, we name the experimental task "target pointing" instead of "target acquisition". We estimate the precision of the pointing as the distribution of end distance to the target. This distribution is translated to an equivalent target width using the computation of ew according to Equation 3. ew yields a continuous estimate of the precision achieved by the participants for a given duration, which allows an accurate measure of the limit of user's precision.

Previous works have controlled the pointing time, using ew as a measure of users' performance [23], or keeping a set of discrete target sizes as in more classical Fitts' paradigm [26]. In both cases, the limit of pointing precision was not the scope, and subjects were aware of the duration available for pointing. Similarly to these approaches, in pilot studies, we trained participants to the duration of the trials with a visual and auditory metronome [26]. We observed that participants reduced their speed of motion in order to reach the target exactly at the end of the trial and minimize corrective motions. We removed this bias in our protocol by insuring that the allocated duration to reach the target is unknown to participants, in contrast with previous approaches. This was done by randomizing the presentation order of trial durations. The final protocol includes the following steps:

- Participants move the screen pointer within a *start area* shown on screen as a grey region (Figure 6, left). The start area has a large width to make it an easy target.
- As the pointer is moved, the location of the *next target* is shown as a thin grey line, offset from the pointer by the *amplitude* of the next trial. Thus, the next target moves with the pointer until the pointing task actually starts, which helps participants to plan their pointing gesture.
- The pointing gesture starts when the pointer has remained within the start area for some time set by the system. The time is randomly chosen within the range 500 – 1500ms. This prevents anticipation from the participants.
- When the pointing task starts, the start area is hidden and the moving thin grey line of the next target is replaced by a fixed thick red line (Figure 6, right).
- Participants move the pointer as close as possible to the target. The duration of target presentation is set by the system

and unknown to the participants who are induced to rush to the target and then try to stabilize the pointer.

- The target is hidden at the end of the trial duration, the distance to the target (error) and the distance from the starting point (trial amplitude) are recorded. The next trial is initiated by showing the start area again.

We predicted that the longer the pointing duration, the better the pointing precision achieved by the participants. However, this linear relationship should only apply until a duration that is long enough for the participants to reach their limit of precision. From this duration and above, we should not see any improvement in precision. As a result, when plotting ew against pointing durations, we should see a curve that decreases until it reaches a plateau. The ew associated with the plateau will represent the limit of users' precision.

As the precision *limit* is only related to the target width, not to the distance to target, Berard et al. [4] used a single amplitude in their protocol. Here, we used two amplitudes in order to introduce an additional evidence that participants have reached their precision limit. For short durations, the small amplitude should yield better precision than the large one because of the extra time provided by the smaller distance to cover. This extra time should serve to stabilize the pointer on the target. However, once the pointing duration is long enough for participants to reach their precision limit, the extra time should be of no use. As a result, when plotting ew against pointing durations for both amplitudes, we should see the curve of the larger amplitude *above* the one of the smaller amplitude, until they converge at the precision limit.

The experimental protocol focuses on the limit of pointing precision, not on pointing in the general case. Consequently, the amplitudes should be chosen small enough such as they only require the *final* pointing gesture from the participants. If pointing with a mouse for example, they should not require clutching. In the case of linear touch pointing, amplitudes must be chosen so that the pointing can be performed in the common precision pointing way: by a small deformation of the finger on the surface, without actual sliding of the finger. For rotational pointing, amplitudes must be small enough so that the rotation of the finger and the wrist can be performed in a single gesture, without clutching.

In order to implement this experimental protocol, we needed to measure finger motions with great precision. We developed two optical finger trackers for this.

HIGH PRECISION OPTICAL FINGER TRACKING

Measuring the limit of user precision is a task that has very high requirements in term of the precision of the input device. Devices are usually designed to support Human-computer interaction. As such, they only need to provide a precision that is similar to what users can achieve. Consequently, we were not aware of any commercial or experimental touch-sensing device that could satisfy the requirements of a *measuring* device. We built two devices for the measurement of finger motion especially designed to optimize precision. Our experiments also required that the devices had low latency, as lag is well known to degrade Human performance [16], and that

they would be robust enough to run without failure during the experimental sessions. These demanding requirements were counterbalanced by the low requirements on the range of motion and by the possibility of operator supervision. Optical sensing was a natural choice because of the flexibility it offers in term of tracking approaches and parameters.

Image sensing

For image sensing, we used an AVT Marlin F-131B camera with a 12mm C-Mount lens. The camera has a CMOS sensor of resolution 1280x1024 that has high sensitivity in the near infrared (IR) spectrum. In order to have a strong control on illumination, we chose to work in the IR spectrum: a filter was mounted on the camera to block the visible light spectrum, and a RayLed RayMax 25 infrared illuminator was used to light the finger.

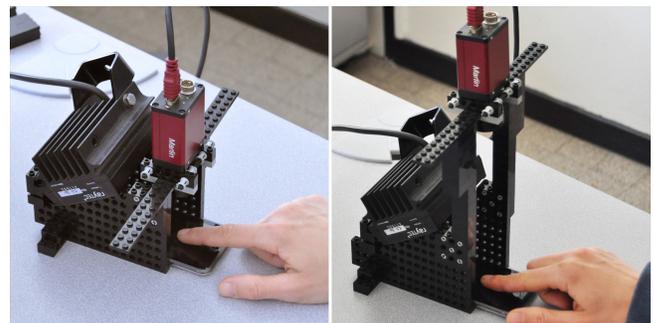


Figure 1. Camera setups for linear tracking (left) and rotational tracking (right).

The camera was mounted vertically above a touch device using a custom made frame built from Lego™ blocks, as shown in Figure 1. The frame also supported the IR illuminator oriented towards the finger. The close proximity of the light source allowed the use of a short exposure time on the camera, a requirement for high frame rates. The touch device was only used as a compliant touch surface and was not switched on. The frame had two different configurations allowing different fields of view for the camera. When measuring the finger linear position, the tip of the finger was tracked on a very limited range. The camera was set very close to the finger to increase measurement precision. There was a minimal distance, though, where the image became too blurry for correct tracking. We found this distance at 60mm between the lens and the touch surface. For the measurement of the finger rotations, a larger field of view was needed to see the sides of the finger. We set the lens at 175mm from the touch surface.

CMOS sensors have the ability to provide the reading of only a subpart of the sensor. This “Region Of Interest” (ROI) is set by sending commands to the camera. This gives great flexibility in choosing a tradeoff between the size of the captured image and the frame rate. We used two different settings for linear and rotational tracking. For linear tracking, we only needed mono-dimensional tracking in the left-right direction. The ROI was thus set to 600×300 pixels, which allowed a frame rate of 83Hz. The rotational tracking required a more square ROI, we set it to 500×400 pixels which translated to a frame rate of 78Hz.

The video signal was processed by an Apple Mac Pro computer equipped with 2 quad-core 3 GHz Xeon processors. We made sure that no buffering occurred in the video acquisition pipeline in order to minimize latency. Tracking was mono-thread for the linear tracking, but was parallelized for the rotational tracking as detailed below.

Tracking linear finger motion

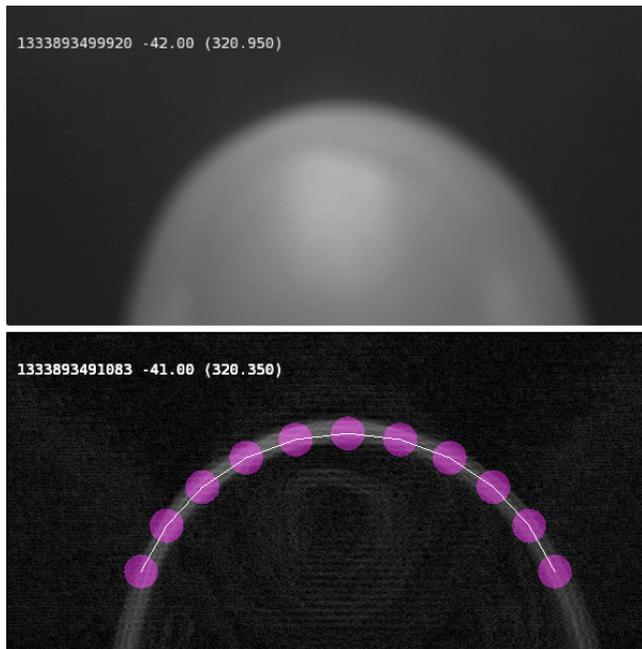


Figure 2. Linear tracking. The out of focus input image of the tip of a finger (top). The arc of a circle model with 11 control points, fit to the Sobel edge image (bottom).

The Arc model

The main objective of our tracker was to achieve high *precision*. With our setup, the 600 pixels wide image covered 25mm of the surface, hence the measurement at the pixel level was roughly 600dpi (dot per inch). But video signals are known to be noisy at the pixel level, reducing the measurement precision if noise is not handled properly. Moreover, pilot studies revealed that 600dpi would not be enough for an accurate measurement of user’s precision. We dealt with the noise problem, and the need for higher precision, by fitting a simple model to the image. The model was an arc of a circle with an angle of 135°. It was fitted to the image through 11 control points, as illustrated in Figure 2 (bottom). Each control point had a mask of size 33x33 pixels with values defined by a 2 dimensional Gaussian with a standard deviation of 16 and centered in the mask.

To fit the model to the image, we first processed the greyscale input image by a Sobel filter [10] to generate an edge image (the background in Figure 2, bottom). We defined the “score” of the model, at any (x,y) location, as the sum of the scores of each of its control points. The score of a control point is the sum of the edge image values at locations covered by the control point’s mask, weighted by the mask’s Gaussian coefficients. The model was fitted to a new frame by an exhaustive

search of the maximum score for all possible locations in the 2D neighborhood of the model’s previous position.

In order to cope with various finger sizes, the radius of the model was adapted by the operator for every new participant of the experiment. Still, an arc of a circle is only a rough approximation of the shape of a finger, especially when the finger is rotated along its own axis, for example to reach the edges of the work surface. We observed, however, that the arc model allowed our tracking to be very stable, even when the fingertips were not circular anymore. Here, we should emphasize that the tracker did not need to be *accurate* in estimating the shape of the finger. It was only required to be stable and precise such that participants could make precise corrections when aligning their cursor to a small target.

Sub pixel tracking

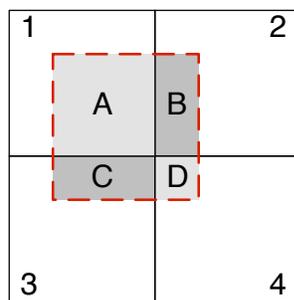


Figure 3. Sub pixel score computation for one particular mask pixel. In the general case, a mask pixel (thick dashed red square) intersects 4 image pixels (numbered thin black square). The contribution of this mask pixel to its control point’s score is the sum of the products of the mask pixel Gaussian value by the image pixel values, weighted by the surface of the area of intersection (labeled A to D here).

The first benefit of the model is to handle noise at the pixel level: the score of the model depends on 11 control points, each of those points depending on 33x33 pixels. Hence the global position estimation averages the uncorrelated noise of 11969 pixels. The second benefit of the model is to allow the measurement of sub-pixel accuracy: control point score computation was implemented for any floating point position in the image, as illustrated in Figure 3. The model’s maximum search was thus performed on a grid of 1/2 pixel. In order to maintain computation time compatible with the input frame rate, we optimized the search by making it hierarchical: the search was first performed on a grid of 9x9 (horizontal and vertical) locations separated by 4 pixels and centered on the model location on the previous frame. We divided the grid spacing by 2 at each level, resulting in only 3x3 locations having to be searched around the maximum location found at the previous level. The search was thus performed on 3 levels in order to reach a grid spacing of 1/2 pixel. To get confidence that this approach did not miss any local maximum, we plotted a set of score maps at the 1/2 pixel grid spacing on a 2 x 2 pixel wide area around the found maximum. We observed that the maps had a smooth dome-shaped surface with a single maximum. Finally, the raw output of the tracker was multiplied by 2, so that the tracker outputted integer number of “ticks”.

We verified that the first level of search, in a 36×36 pixel wide area, combined with the $83Hz$ frame rate, was large enough to track very fast finger motions. During the experiments, we never observed any tracking failure. Tracking only had to be bootstrapped when the finger first appeared in the image: at the beginning of a session, or after participants had removed their finger during pauses. Bootstrapping was performed by an automatic algorithm that found the topmost line in the edge image having a sum of pixel greater than a threshold, and then initializing the model at the barycenter of this line.

Evaluation of stability and precision

As a tool for measuring precision, instead of a tool for an interactive system, our tracker has a specific set of requirements. Namely, its output must be perfectly stable, and its precision has to be accurately measured.

For static stability, we cut a piece of cardboard in the shape of a finger. We then simply put the cardboard finger in several different locations under the camera, and checked that a perfectly stable object yielded a perfectly stable output from the tracker. For dynamic stability, we recorded the output of a finger moving at roughly constant slow speed and verified that the trajectory was regular and spanning all tick positions.

To evaluate the precision, we used the cardboard finger again, this time to point at two different graduations, as far as possible of each other, on a ruler put on the touch surface. We used the tracker control window to align the center of the model to the two graduations and to read the tracker’s reported position. A physical distance of $22mm$ yielded a distance of 1040 ticks of the tracker. The tracker precision was thus evaluated at $1040 \times 25.4/22 = 1200dpi$. We estimated that our measurement error was within $0.5mm$ on both positions, hence we estimated the physical distance in the range $21 - 23mm$ and the tracker’s precision in the range $1150 - 1260dpi$.

Tracking rotational finger motion

The Snake Model

The arc of circle model was good enough to extract the position of the fingertip, but not to extract the orientation of the finger because it did not track the sides of the finger. We thus used a snake [13] model in order to track more of the participants’ fingers, while adapting to their various shapes. The snake model shared its structure with the circular arc: it consisted in a set of control points connected with each other at fixed distance. As the *scale* of the scene was smaller (in order to see more of the finger), the control points were smaller too: their mask had a size of 9×9 pixels with coefficients from a 2D Gaussian of standard deviation 4. The model fitted to a finger is shown on Figure 4 (right).

The model was initialized by the operator clicking on the center of the tip of the finger while in vertical orientation. From this point on, the system found the control points on the edge of the finger as detailed in Figure 5. For increased precision, we used the same sub-pixel score maximization approach as with the arc model.

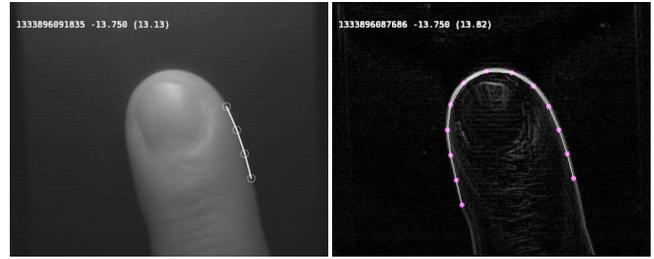


Figure 4. Rotational tracking. (right) The snake model with 13 control points, fit to the Sobel edge image. (left) The finger’s orientation is computed as the mean of the 3 orientations defined by the 4 last controlled point. Only the well-lit side is used.

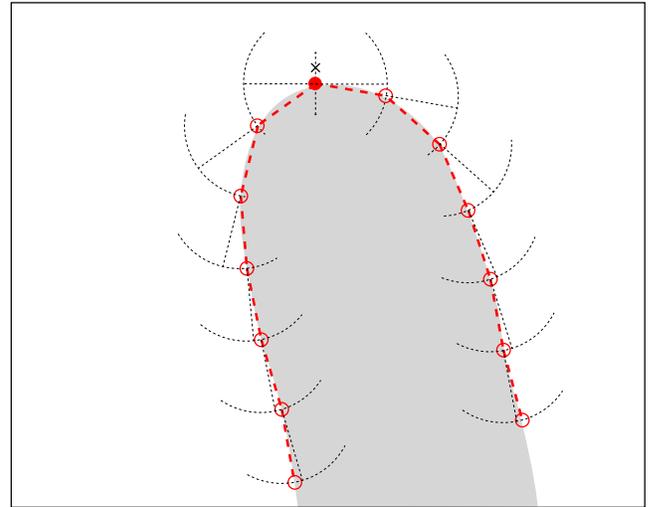


Figure 5. Defining the snake shape (thick dashed red line) on the edge of the finger (grey shape). The operator clicked on the “x”. The first “anchor” control point (red disk) is searched on a vertical neighborhood. The other control points, on each side of the anchor (red circles), are searched at 40 pixels, on 90° arcs oriented from the two previous control points, except for the 2 first which are search at orientations 0 and 180° .

Once the model had been initialized, its shape was *frozen* and it was only allowed three degrees of freedom: (x, y, θ) , θ being the rotation in the surface plane. The snake’s U shape had to be frozen to prevent it to drift along the finger edges, which would prevent the reliable tracking of (x, y) , and ultimately of θ .

Tracking

Tracking was performed, as with the arc model, by an exhaustive search in the neighborhood of the previously found position. The neighborhood was now 3-dimensional, which significantly increased computation cost. However, the search parameter space was easily split and distributed on the 8 cores of the computer. We used the same hierarchical approach, this time with 3 levels, to optimize the search as with the arc model. Initial search was performed on a grid of 16×16 locations spaced by 2 pixels, and 22° of angle by steps of 1.15° . At the third level, the resolution of the search was 0.1 pixels in position and 0.06° in rotation. Here again, the wide search area of the initial level, combined with the $78Hz$ frame rate, allowed the tracker to never fail during the experiment.

We first planned to use the θ angle found in the tracking as the orientation of the finger. But we observed that this fixed snake model would not be stable enough for our experiments: using the cardboard finger, we computed the standard deviation of the output for a perfectly stable finger, and found that it would not allow a stable output below 1° . We attributed this problem to the inability of the fixed snake to adapt to slight shape variations along the finger edges during large rotations. Orientation estimation was thus implemented in two stages. In the first stage, the fixed snake was searched as described above. In a second stage, the shape constraint was released and the side control points were searched using the same approach as shown in Picture 5, only this time the anchor point was defined by the first stage instead of the click of the operator. The role of the fixed snake was thus to reliably track the anchor control point at the tip of the finger, so that an accurate snake could be fit in the second stage.

Once the accurate shape of the finger had been extracted, we computed the finger's orientation from the last control points at the end of each side of the snake. On each side, we averaged the 3 orientations of the segments defined by the 4 last control points, as shown in Figure 4 (left). We initially averaged the two resulting orientations from both sides. We observed however that on large rotations, due to the light source coming from the top of the image, the side of the finger closer to the bottom of the image was in the shadow and could become unstable. We thus simply estimated the finger orientation from the well lit side of the finger.

Precision

Using the cardboard finger and plotting the output of rotating fingers, we found that setting a threshold at 0.25° before reporting a new rotation produced a stable and continuous output. We thus estimated the rotational tracker's resolution at 0.25° .

EXPERIMENTAL EVALUATION OF FINGER POINTING PRECISION

We used the trackers to run two experiments aimed at measuring the linear and rotational limit of user's precision in finger pointing. We implemented two *duration controlled target pointing* experiments as defined and motivated in the third section of this paper.

Visual feedback

The visual feedback was displayed on a 120Hz Samsung SyncMaster 2233RZ 1680×1050 pixels monitor. The visibility of the pointer and target was improved using a zoom factor of 4 (a 1 tick target was represented by 4 pixels). As Bohan et al. have shown, increasing visual magnification essentially amounts to zoom-in to the precision limits of the effector, after which performance asymptotes[5]. The visual feedback of the experiments is shown on Figure 6.

Experiment design

Twelve volunteers participated in each experiment, aged 32.5 (mean) ± 10.8 (stddev) for the linear experiment and, 29.1 ± 5.53 for the rotational experiment. Six users participated in both experiments. All of the participants had extensive

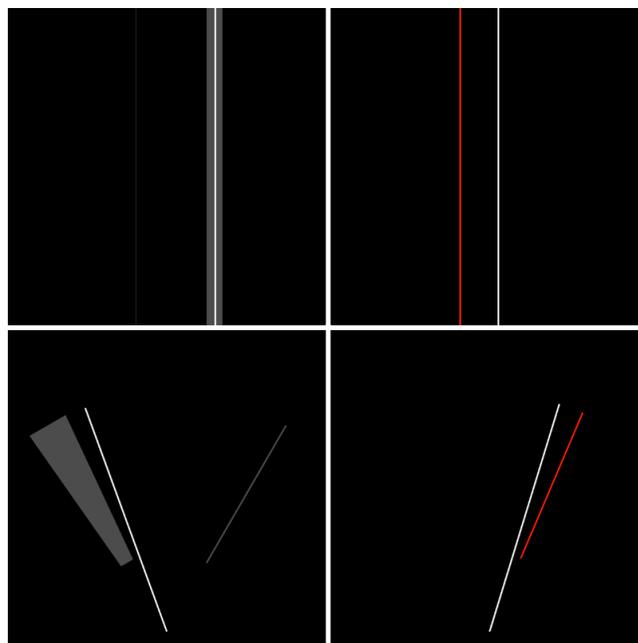


Figure 6. Visual feedback of the linear (top) and rotational (bottom) experiments.

previous experience with laptops' touchpads. Nine of them in each experiment had extensive previous experience with direct touch devices (such as smartphones).

Participants were asked to position their index finger on the touch device, below the camera, in a comfortable posture. Most participants adopted a posture with the finger close to the horizontal, but a few preferred a more vertical posture with the finger at around 45° . Participants were given a few trials to understand the task and to find their most comfortable posture.

Participants were asked to be as precise as possible in pointing to 1 *tick* wide targets at two different amplitudes: (80, 120) ticks and (30, 50) degrees for the linear and the rotational experiment respectively. For the linear experiment, both amplitudes allowed pointing by a *slight roll of the finger* leading to a slight deformation, i.e. without any sliding. For the rotational experiment, both amplitudes required a significant motion of the palm of the pointing hand.

Targets were displayed with 8 different durations: (800, 900, 1000, 1100, 1200, 1400, 1600, 2000) ms. All combinations of amplitude and durations were presented 5 times within a block, in random order, with different orders in each block. Directions (left/right, clockwise/counterclockwise) were reversed after each trial. Participants performed 6 blocks with a 1 min break between blocks. The total number of trials was 5760 for each experiment and each experimental sessions lasted 1/2h.

Data processing

For each group (subject X amplitude X duration combination, 30 trials) we rejected trials for which the error was greater than 3 times the error mean of the group (in total, 2.3% of

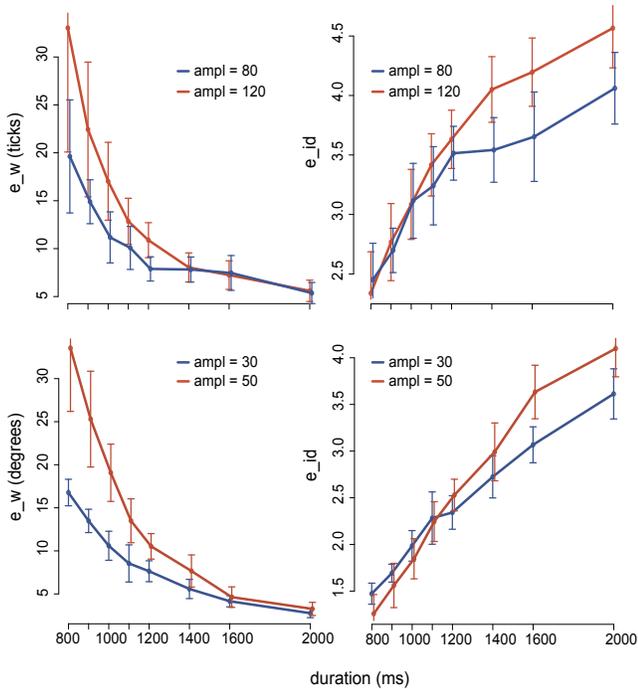


Figure 7. Result of the linear experiment (top) and rotational experiment (bottom). Average effective width (left) and effective ID (right) achieved by all participants as a function of the trial duration, with 95% confidence intervals, for amplitude 80 and 120 ticks (top) and 30 and 50 degrees (bottom).

the trials for the linear experiment, 1.3% for the rotational experiment). We then computed two parameters: the effective target width (ew), according to Equation 3; and the effective index of difficulty $eid = \log_2(ea/ew + 1)$, with ea being the effective amplitude, computed as the average of the pointing amplitude.

Results

Results of the two experiments are summarized in Table 1 and illustrated in Figure 7. Distances are expressed in ticks of our 1200dpi linear tracker ($1tick = 2.12 \times 10^{-2}mm$).

The effects of Amplitude and Duration on the effective width (ew) were tested for each experiment using 2-ways within subjects Analyses of Variance (ANOVA). Post-hoc comparisons of ew and eid between amplitude were run for the 8 levels of duration using paired t-tests with Bonferroni correction (testing using $8 \times p$).

Linear experiment

In the linear experiment, between 800ms and 1200ms, ew decreases from 20 tick to 8 tick for Amplitude 80 tick and from 33 tick to 11 tick for Amplitude 120 tick. For both amplitudes, ew stabilizes at about 8 tick between 1400ms and 1600ms, but then improves again to reach about 5 tick at 2000ms (Figure 7). ANOVA shows that ew changes with Duration [$F(7, 77)=23.7, p<.0001$] and with Amplitude [$F(1, 11)=14.4, p<.01$]. The effect of Amplitude also interacts with the effect of Duration [$F(7,77)=7.59, p<.0001$]. Post-hoc comparisons show significant differences in ew between

Table 1. Results of the experiment. Average on all participants of the effective width (ew) and effective ID (eid) across durations, for the two amplitudes and the two experiments (top and bottom). Amplitude (A) and ew are expressed in ticks for the linear experiment and in degrees for the rotational experiment.

duration (ms)	Linear experiment					
	ew		eid			
	A=80	A=120	A=80	A=120		
800	19.62	33.03	*	2.45	2.34	
900	14.89	22.43		2.70	2.77	
1000	11.17	17.03	*	3.11	3.08	
1100	10.08	12.85		3.24	3.42	
1200	7.88	10.88	**	3.51	3.63	
1400	7.82	8.04		3.54	4.05	*
1600	7.46	7.24		3.65	4.20	*
2000	5.36	5.60		4.06	4.57	**
duration (ms)	Rotational experiment					
	ew		eid			
	A=30	A=50	A=30	A=50		
800	16.77	33.55	**	1.47	1.26	
900	13.48	25.30	**	1.69	1.56	
1000	10.59	19.07	**	1.98	1.85	
1100	8.54	13.50	**	2.28	2.24	
1200	7.64	10.53	**	2.34	2.53	
1400	5.58	7.66		2.72	2.99	
1600	4.15	4.63		3.07	3.63	**
2000	2.76	3.26		3.61	4.10	*

* $8p < .05$, ** $8p < .01$,
post-hoc comparisons between amplitude

the two Amplitudes for durations smaller than 1200ms but not for longer durations.

Similar eid are observed for both amplitudes for durations from 800ms to 1200ms. For durations longer than 1200ms, the improvement of eid with duration then changes according to the Amplitude, from 3.54 at 1400ms to 4.06 at 2000ms for Amplitude 80 tick and from 4.05 to 4.57 for Amplitude 120 tick. ANOVA shows a significant effect of Amplitude [$F(1,11)=20.97, p<.001$] and Duration [$F(7,77)=95.7, p<.0001$] on eid and a significant interaction of the two factors [$F(7,77)=5.42, p<.0001$]. Post-hoc comparisons show that differences in eid between the two amplitudes are significant only for durations greater than 1200ms.

Rotational experiment

The rotational experiment does not show the same stabilization of eid as with the linear experiment for durations in the range [1200 – 1600ms]: eid appears to improve regularly up to the longest duration of 2000ms. However, the convergence of the curves of the 2 amplitudes is still observable (at duration 1600ms). ANOVA shows a significant effect of Amplitude [$F(1,11)=57.6, p<.0001$] and Duration [$F(7, 77)=107, p<.0001$] and Amplitude X Duration [$F(7, 77)=19.2, p<.0001$] on ew . Post-Hoc comparisons show that ew is significantly different for Amplitude 50° as compare with Amplitude 30° only for durations shorter than 1400ms.

ANOVA shows a significant effect of Amplitude [$F(1,11)=8.04, p<.05$], Duration [$F(7,77)=247, p<.0001$] and Amplitude X Duration [$F(7,77)=8.84, p<.0001$] on eid . eid linearly increases with the duration and similar values are observed between 800ms and 1400ms, the difference being only significant at 1600ms and 2000ms.

Discussion

Our users' studies were a first effort to evaluate the user resolution of finger pointing on a touch surface. Our results are partially consistent with our expectations, as threshold in precision is not as clear as predicted, especially for the rotational pointing. However, our results provide a first objective measurement of users' precision.

Linear experiment

In the linear experiment, we observe a stabilization of users' performance at about 8 *tick* for durations between 1200ms and 1600ms: for Amplitude 80 *tick*, this 400ms increase in duration only yields a precision improvement of 0.42 *tick*, while the previous 400ms duration increase (from 800ms to 1200ms) yields an improvement of 11.74 *tick*. Also, at duration 1400ms and above, the effect of Amplitude seems to disappear as the achieved *ew* is similar for both amplitudes. Hence, around 1400ms, users' performances move away from Fitts' Law prediction. Consistently with this observation made from *ew*, *eid* curves for the two Amplitudes overlap for duration smaller or equal to 1200ms, as predicted by Fitts' law. The two curves then diverge starting at 1400ms, as the increase in duration doesn't improve user's precision anymore for the smaller Amplitude. In addition, the slope of *eid* curves for the two amplitudes are less steep after 1400ms, indicating that participants become substantially less efficient than with lower durations. Using the tracker's precision evaluated at around 1200dpi (see the above section "Evaluation of stability and precision"), the 8 *tick* translate to a precision of $1200/8 = 150dpi$. In other words, users can point at targets as small as 0.17mm within 1.4 sec, for amplitudes of movement up to 120 *tick* (2.54mm).

The surprising outcome of the linear experiment is that precision can still be improved *after ew* has stabilized: when the duration increased past 1600ms to 2000ms, *ew* improved by about 2 *tick* for Amplitude 80 *tick* and by about 1.6 *tick* for Amplitude 120 *tick*. With the highest amount of time and lowest amplitude that they were given, our participants were able to point at targets as small as 0.11mm (equivalent to 224dpi). Our experiment indicates that even though there is a threshold in precision achievable in a "regular" amount of time (i.e. predictable by Fitts' Law), when the duration available to reach the target increases, users probably enter another mode of control. This mode only materialize for long pointing durations, where the time for initial approach to target is minor compared to the time for final adjustments, hence the effect of Amplitude disappears in the achieved pointing precision. However, due to the large amount of time required, this mode of operation might not be suitable for frequent interactions such as positioning a cursor or acquiring small buttons in a dense display.

Rotational experiment

The rotational experiment does not show the same profile as the linear experiment. Rather than stabilizing at some precision level, the curves *ew* for the two amplitudes show a regular improvement up to the last duration (Figure 7, bottom). However, a divergence of the *eid* curve is here again visible, as indicated by the significant differences of *eid* for

the two amplitudes at Duration 1600ms and above. The lack of precision stabilization in our rotational experiment could be attributed to the fact that the two angular amplitudes used in this experiment may have been too high, inducing the major part of the pointing time to be allocated to the approach of the target, and so, not enough time for final adjustments. This point to the necessity to investigate the effect for smaller amplitudes.

However, as a first attempt to evaluate user's precision in rotational pointing, our study provides a guideline for the development of interaction on touch surface using rotational movement of the finger. Hence on Table 1, designer can find the user's precision that can be expected depending on the duration and on the amplitude of the pointing. To our knowledge, this constitutes the first empirical evidence that users can be quite precise in rotational pointing, down to 3.26° for duration of 2 sec and Amplitude 50°. This indicates, for example, that reliable selection of 1 sector among 15 in a 50° pie menu could be performed by direct pointing.

CONCLUSION

In this paper, we contributed with the design and evaluation of two high precision optical finger trackers, one for linear motions and one for rotations. The trackers run at frame rates above 70Hz, they provide robust and stable outputs, and their precisions were evaluated at 1200dpi for the linear tracker and 0.25° for the rotational tracker. These characteristics make them valuable experimental tools that could be useful to address other properties of finger gestures and to evaluate interactions that rely on precise finger motions. In addition, our trackers open the way to improvements in their tracking range and autonomy, which would have direct applications in the design of interactive systems.

The main objective of this work was to estimate the limit of users' precision in linear and rotational touch pointing. We found this limit at 150dpi, or 0.17mm for linear pointing. It appears that we did not reach the limit for rotational pointing, and additional research will be required here to decide if there is a threshold. Still, we discovered that for rotational pointing, users can be quite precise, achieving for example 2.76° of precision for a 2s pointing duration and 30° of amplitude. This result should motivate the design of new interactions based on precise pointing, and the design of new devices that are able to sense this level of precision in rotational pointing.

ACKNOWLEDGMENTS

The authors wish to thank Renaud Blanch for his contributions to this work.

REFERENCES

1. Pär-Anders Albinsson and Shumin Zhai. High precision touch screen interaction. In *ACM Conference on Human Factors in Computing Systems (CHI)*, pages 105–112. ACM, 2003.
2. Patrick Baudisch and Gerry Chu. Back-of-device interaction allows creating very small touch devices. In

- ACM Conference on Human Factors in Computing Systems (CHI)*, pages 1923–1932. ACM, 2009.
3. Hrvoje Benko, Andrew D. Wilson, and Patrick Baudisch. Precise selection techniques for multi-touch screens. In *ACM Conference on Human Factors in Computing Systems (CHI)*, pages 1263–1272. ACM, 2006.
 4. François Bérard, Guangyu Wang, and Jeremy R. Cooperstock. On the limits of the human motor control precision: the search for a device’s human resolution. In *IFIP Conference on Human-Computer Interaction, INTERACT*, pages 107–122. Springer-Verlag, 2011.
 5. Michael Bohan, Daniel S. McConnell, Alex Chaparro, and Shelby G. Thompson. The effects of visual magnification and physical movement scale on the manipulation of a tool with indirect vision. *Journal of Experimental Psychology: Applied*, 16(1):33–44, March 2010.
 6. Géry Casiez, Daniel Vogel, Ravin Balakrishnan, and Andy Cockburn. The impact of control-display gain on user performance in pointing tasks. *Human-Computer Interaction*, 23(3):215–250, 2008.
 7. Olivier Chapuis and Pierre Dragicevic. Effects of motor scale, visual scale and quantization on small target acquisition difficulty. *ACM Trans. Comput.-Hum. Interact.*, 18(3), 2011.
 8. Ali Erol, George Bebis, Mircea Nicolescu, Richard D. Boyle, and Xander Twombly. Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, 108(1–2):52 – 73, 2007. Special Issue on Vision for Human-Computer Interaction.
 9. Paul M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6):381–391, 1954.
 10. Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2001.
 11. Christian Holz and Patrick Baudisch. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. In *ACM Conference on Human Factors in Computing Systems (CHI)*, CHI ’10, pages 581–590, New York, NY, USA, 2010. ACM.
 12. Christian Holz and Patrick Baudisch. Understanding touch. In *ACM Conference on Human Factors in Computing Systems (CHI)*, CHI ’11, pages 2501–2510, New York, NY, USA, 2011. ACM.
 13. Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1988.
 14. Julien Letessier and Francois Berard. Visual tracking of bare fingers for interactive surfaces. In *ACM Symposium on User Interface Software and Technology (UIST)*, pages 119–122, 2004.
 15. I Scott MacKenzie. Fitts’ law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 7:91–139, 1992.
 16. I Scott MacKenzie and Colin Ware. Lag as a determinant of human performance in interactive systems. In *ACM Conference on Human Factors in Computing Systems (CHI)*, pages 488–493, 1993.
 17. Shahzad Malik and Joe Laszlo. Visual touchpad: a two-handed gestural input device. In *ACM International Conference on Multimodal Interfaces (ICMI)*, pages 289–296. ACM, 2004.
 18. K. Oka, Y. Sato, and H. Koike. Real-time fingertip tracking and gesture recognition. *Computer Graphics and Applications, IEEE*, 22(6):64–71, Nov/Dec 2002.
 19. Mahfuz Rahman, Sean Gustafson, Pourang Irani, and Sriram Subramanian. Tilt techniques: investigating the dexterity of wrist-based input. In *ACM Conference on Human Factors in Computing Systems (CHI)*, pages 1943–1952. ACM, 2009.
 20. Simon Rogers, John Williamson, Craig Stewart, and Roderick Murray-Smith. Anglepose: robust, precise capacitive touch tracking via 3d orientation estimation. In *ACM Conference on Human Factors in Computing Systems (CHI)*, pages 2575–2584. ACM, 2011.
 21. Anne Roudaut, Stéphane Huot, and Eric Lecolinet. Taptap and magstick: improving one-handed target acquisition on small touch-screens. In *Proceedings of the working conference on Advanced visual interfaces, AVI ’08*, pages 146–153, New York, NY, USA, 2008. ACM.
 22. Nicolas Roussel, Géry Casiez, Jonathan Aceituno, and Daniel Vogel. Giving a hand to the eyes: Leveraging input accuracy for subpixel interaction. In *ACM Symposium on User Interface Software and Technology (UIST)*. ACM, 2012.
 23. Richard A. Schmidt, Howard Zelaznik, Brian Hawkins, James S. Frank, and John T. Quinn Jr. Motor-output variability: A theory for the accuracy of rapid motor acts. *Psychological Review*, 86(5):415–451, 1979.
 24. Daniel Vogel and Patrick Baudisch. Shift: a technique for operating pen-based interfaces using touch. In *ACM Conference on Human Factors in Computing Systems (CHI)*, pages 657–666. ACM, 2007.
 25. Feng Wang and Xiangshi Ren. Empirical evaluation for finger input properties in multi-touch interaction. In *ACM Conference on Human Factors in Computing Systems (CHI)*, pages 1063–1072. ACM, 2009.
 26. Jacob O. Wobbrock, Edward Cutrell, Susumu Harada, and I. Scott MacKenzie. An error model for pointing based on fitts’ law. In *ACM Conference on Human Factors in Computing Systems (CHI)*, pages 1613–1622. ACM, 2008.