



**HAL**  
open science

## Congruent Indirect Touch vs. mouse pointing performance

François Bérard

► **To cite this version:**

François Bérard. Congruent Indirect Touch vs. mouse pointing performance. International Journal of Human-Computer Studies, 2024, 187, pp.103261. 10.1016/j.ijhcs.2024.103261 . hal-04522270

**HAL Id: hal-04522270**

**<https://hal.science/hal-04522270v1>**

Submitted on 2 Apr 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



## Congruent Indirect Touch vs. mouse pointing performance

François Bérard

Université de Grenoble-Alpes/Grenoble-INP, LIG, 700 av centrale, 38400, Saint Martin d'Hères, France

### ARTICLE INFO

#### Keywords:

Touch  
Congruent Indirect Touch  
Pointing performance  
Mouse

### ABSTRACT

We study Congruent Indirect Touch (CIT) interaction in a desktop context. CIT only differs from direct touch in that the motor and display spaces are separated: touch occurs on the horizontal desk; while users' visual focus is on a vertical display where small pointers show the position of the fingers. We introduce an accurate fingertip tracking approach based on optical tracking and fingertip modeling as a sphere. This allows updating the pointer when the finger is hovering above the surface and implementing an efficient CIT interaction. This interaction was evaluated in a longitudinal user study. Six participants with no experience with CIT performed target acquisitions on eight different days. Throughput was measured with CIT and mouse. In the last session, two participants had similar throughput with both interactions; the four others were notably more efficient with CIT. Averaged across all participants, throughput improved by 14%. This study promotes the study of CIT as a potential efficient replacement for the mouse on the desktop.

### 1. Introduction

Two different spatial pointing interactions techniques dominate HCI. On the desktop, pointing is mostly performed with a mouse or a touchpad. It is an indirect, relative control using a speed-dependent control-display gain. On mobile devices, interaction is mostly performed with direct touch: an absolute position control that entails a constant control-display gain equal to 1. While desktop pointing is suitable for long work sessions and allows to acquire smaller targets, direct touch tapping has been observed as being more efficient than mouse pointing for large targets (Cockburn et al., 2012; Forlines et al., 2007; Sasangohar et al., 2009; Sears and Shneiderman, 1991). Furthermore, direct-touch devices support interactions with multiple streams of spatial input such as two-handed interaction.

In this work, we contemplate merging these two forms of interaction to get the best of both worlds: the efficiency and multi-stream ability of direct touch with the precision and comfort of desktop pointing. Direct-touch is readily available on the desktop with the use of a multitouch display and an operating system that supports multiple streams of spatial input, such as Microsoft Windows. However, as in mobile pointing, the imprecision of direct-touch known as the “fat finger problem” (Holz and Baudisch, 2010) remains: the finger hides the target and the finger-surface contact area creates an ambiguity on the precise location of the pointing. In addition, fatigue to the arms known as the “gorilla-arm-effect” occurs when the touch display is set up vertically (Boring et al., 2009). If it is set up horizontally, pain occurs to the neck as the head constantly looks down. Tilted displays only allow to control the arm/neck pain tradeoff, but do not offer the

optimal ergonomics of the desktop: arms resting on the horizontal table, head looking straight to a vertical display.

To bring to the desktop the benefits of direct touch interaction without its shortcomings, we turn to a seldom studied form of interaction: Congruent Indirect Touch (CIT). In geometry, congruent shapes coincide exactly when superimposed. With CIT, as with direct touch, the trajectories of the finger and of the controlled digital object are congruent. CIT only differs from direct touch in that the motor and display spaces are separated: touch occurs on the horizontal desk; while users' visual focus is on a vertical display where small pointers show the position of the fingers.

CIT may look similar to the common touchpad pointing interaction: both involve touching a horizontal surface while looking at a pointer on a vertical display. However, they differ in several key characteristics: touchpad pointing is used with a relative control that involves a dynamic transfer function; while CIT entails absolute control with a constant control-display gain equal to 1. Touchpads require controlling a pointer which is then used to acquire targets; while with CIT targets are immediately acquired by landing a finger on them, as with direct touch. Finger hover tracking is required for CIT to provide visual feedback, but not for touchpads. Still, CIT and touchpad interaction share the benefits of the separation of the motor and display space: ergonomic problems are avoided by using a vertical display and a horizontal motor space. The fat finger problem is avoided because the fingers are not occluding the display and the fingers' positions are represented by small cursors on the display, eliminating the ambiguity of the contact area.

E-mail address: [francois.berard@univ-grenoble-alpes.fr](mailto:francois.berard@univ-grenoble-alpes.fr).

<https://doi.org/10.1016/j.ijhcs.2024.103261>

Received 14 February 2023; Received in revised form 14 March 2024; Accepted 18 March 2024

Available online 21 March 2024

1071-5819/© 2024 The Author. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

While CIT does not suffer from the same limits of direct touch, the effect of separating the motor and display spaces must be assessed: the 3D translation and rotation from desk to display may have a cognitive cost that dissolves the performance benefit of direct touch over mouse pointing. Despite its potential to bring the direct-touch benefits to the desktop, CIT was seldom studied. This may be a consequence of the challenge to implement hover and touch tracking with enough precision, accuracy and responsiveness to sustain efficient user interactions.

Our work contributes to the general aim of making the best use of users' motor control abilities. We study if CIT can sustain higher throughput compared to the mouse, as this was observed with direct touch. We also assess the efficiency of CIT on small targets because they are inefficiently acquired with direct touch. Our work focuses on users' abilities rather than technology, hence we provide our best effort to build a system that reduces the negative effect of the technology on users' performances, regardless of its viability out of the lab. We introduce a novel fingertip tracking approach based on 6 degrees of freedom optical tracking of the fingertip and modeling it as a sphere. The approach is implemented at 165 Hz with 26 ms of end-to-end latency. It achieves stable and precise touch/release detection with a hysteresis of 0.2 mm and pixel-stable positional tracking. The approach can generalize to other motor control performance evaluation that require hover tracking, such as back of the device interactions (Gugenheimer et al., 2016; Wigdor et al., 2007). We use this approach to run a longitudinal experiment on six participants to compare CIT and mouse pointing performances following the ISO9241-9 standard (Soukoreff and MacKenzie, 2004). The experiment reveals that using a mouse initially outperforms CIT, but after eight sessions, CIT outperforms the mouse by 14% on average. CIT performance is comparable to the mouse on difficult targets with small sizes down to 3 mm, but it is notably higher with a 37% improvement on the largest targets. This study reveals that CIT can notably increase users' desktop pointing capabilities compared to the status quo and that it is a practicable interaction for small targets. Pointing at targets being a fundamental and ubiquitous building block of graphical user interactions, this study should put desktop CIT under careful examination.

## 2. Previous work

### 2.1. Performance advantage of direct touch

Direct touch was compared to the mouse in several studies (Cockburn et al., 2012; Sasangohar et al., 2009; Forlines et al., 2007; Sears and Shneiderman, 1991). Various tasks were tested, such as target acquisition by landing a finger on the target (sometimes called "tapping") (Cockburn et al., 2012; Sasangohar et al., 2009; Forlines et al., 2007), docking by dragging the object (Cockburn et al., 2012; Forlines et al., 2007; Sears and Shneiderman, 1991), bi-manual input using two fingers vs two mice (Forlines et al., 2007), etc. The general agreement is that direct touch is not efficient for dragging, probably because of the friction of the finger on the interactive surface (Cockburn et al., 2012; Forlines et al., 2007). Direct touch is more suitable for target acquisitions where the finger hovers above the surface until it lands on the target. Here, direct touch was systematically observed to be faster than the mouse when the target size was big enough, i.e., not significantly smaller than the tip of a finger. Sears and Shneiderman observed a superiority of direct touch for target sizes at or above 6.9×9.0 mm (Sears and Shneiderman, 1991). In Forlines et al.'s study, direct touch was more efficient than the mouse with all target sizes, as the smallest target was 19.2 × 19.2 mm in size (Forlines et al., 2007). On average, direct touch was found to be 15% more efficient than the mouse. Sasangohar, MacKenzie and Scott observed a clear superiority of direct touch over the mouse for all but their smallest target (1-dimensional, 6.5 mm). Overall, the throughput improvement was 41% (Sasangohar

et al., 2009). Cockburn, Ahlström and Gutwin observed a 34% performance improvement, but also noted high error rates (13%–14%) for the smallest target size (5 mm). MacKenzie measured target selection performances on a smartphone. The smallest target was 6.2 mm in size. The author reports that pilots testing with 2.4 mm targets "was deemed untenable". He observed an average pointing throughput of 6.95 bit/s; which he compared to the mouse throughput reported in the literature and found "a performance advantage for touch in the range of 42%–88%". Overall, these studies are consistent in showing the superiority of direct touch pointing performance over the mouse when targets are larger than 6 mm.

### 2.2. Studies on indirect touch

We are aware of a single study on Congruent Indirect Touch (Schmidt et al., 2009). Most efforts were oriented towards a closely related form of interaction where the trajectory of the finger in motor space is amplified on the display (i.e. the control-display gain is greater than 1). We refer to this as Similar Indirect Touch (SIT): in geometry, if two shapes are similar, each is congruent to the result of a particular uniform scaling of the other.

Malik, Ranjan, and Balakrishnan experimented with SIT to interact with a large display (Malik et al., 2005). Tracking of the hovering hands was performed by Computer Vision (CV) processing of the output of two cameras set above the workspace. Users could interact with the display using the non-dominant hand to position a "workspace" anywhere on the display with low granularity, and using the dominant hand to perform more precise pointing within this workspace. Kim et al. improved on the overhead stereo camera approach and experimented with several indirect touch applications such as 3D object layout and multitouch navigation on a 2D map (Kim et al., 2014). In both studies, the objectives were oriented towards the exploration of novel interaction techniques rather than the measurement of pointing performances.

In 2005, a company named Tactiva developed Tactapad: a desktop SIT device (Tactiva, 2005). The device was made of a large touchpad and a single video camera set above the touchpad. Semitransparent silhouettes were extracted from the camera feed and shown on the display. Demonstrations on Mac OS X illustrated the possibility to activate objects with various fingers. The transparent silhouettes provided positional information of the hovering fingers while avoiding occlusion, however they were notably larger than many targets and suffered from the ambiguity of the touch location as in direct touch.

Schmidt, Block, and Gellersen compared the performance of direct touch vs. CIT on large interactive surfaces (Schmidt et al., 2009). As with Tactapad, a single video camera was set on top of the touch surface. It was used to show contours of the hand on the display and suffered from the same ambiguity as Tactapad. The authors measured that the acquisition of two handles was almost 3 times slower in CIT than in direct touch. This was attributed to the difficulty of participants to coordinate their hands with the contours on the screen, and their difficulty to perceive the fingers' altitude because they were standing in front of the touch surface with their visual focus on the display. The authors suggested to allow users to rest their hands on the surface to avoid the discomfort with hovering over the surface.

Using a combination of depth cameras, Efanov and Lanir experimented with various representation of the hands in the context of desktop CIT/SIT (Efanov and Lanir, 2015). They hypothesized that "an image of the 3D hand contour or skeleton might provide better help to the user than just the fingertips".

Standard capacitive surfaces on smartphones and trackpads report the absolute position of contacts. It is thus straightforward to implement a *blind* variant of CIT or SIT, i.e. with no feedback while hovering. However, when aiming at a particular target on the display, users have to guess where to touch the surface; which yields low accuracy. To compensate for this low accuracy, McCallum and Irani combined

blind SIT for quick translations over large distances with the common trackpad interaction for fine cursor control. This came at the cost of a two-stage target acquisition for small targets (McCallum and Irani, 2009). Gilliot, Casiez, and Roussel quantified the accuracy of blind SIT (Gilliot et al., 2014). In one condition, participants could not look at their hands nor the touch surface: they only relied on proprioception to do the acquisition. In this condition, they measured that a target size of 22.3 mm could be acquired with a 95% success rate when using the smaller input surface. This study provided quantitative measures of the contribution of proprioception in target acquisitions.

Using a digitizing tablet and a pen, rather than a finger, Causse et al. compared the accuracy of indirect pointing with and without hovering graphical feedback (Causse et al., 2014). A marker on the tip of the pen was tracked by a camera set above the surface to provide the graphical feedback while hovering. The group with graphical feedback was twice as accurate as the group with no graphical feedback (5.4 mm vs. 10.8 mm of average distance to the center of the target). While the authors emphasized the benefit of visual guidance in indirect touch, they also observed that aiming movements tended to take longer when participants relied on the visual feedback.

### 2.3. Finger tracking above the surface and touch detection

Tracking fingers to implement CIT/SIT interactions strongly relates to tracking fingers in free-space and detecting when they touch the surface. This has generated a large body of literature in the CV and HCI research fields. Ahmad, Migniot, and Dipanda for example recently surveyed CV techniques for tracking the hand and fingers while interacting with objects (Ahmad et al., 2019). However, the requirements for our study are demanding as we are focusing on users' abilities: the tracking system must have good enough latency, update rate, precision, accuracy, and stability that it does not hinder participants in their motor control. We anticipate that:

- the 2D position on the surface must be notably more precise than 3 mm, the smallest targets in our experiment,
- the required latency may be as low as 25 ms, as MacKenzie and Ware observed that users' performance degraded at 75 ms but not at 25 ms for indirect pointing with a mouse (MacKenzie and Ware, 1993),
- the hover vs. touch detection (i.e., the altitude measurement) must be very accurate, as detailed below.

The requirement for an accurate hover vs. touch detection is, to some extent, the consequence of our daily use of capacitive touch surface; which are both sensitive and robust: we have come to expect that the slightest touch of an application icon with a finger is registered by the system and triggers the launch of the application. More critically in our experiment, accurate touch detection is a requirement to measure participants' maximum input bandwidth. Any inaccuracy may require users to press a little stronger than expected to avoid mis-detection, or to hover a little higher to avoid spurious activation. In both cases, the inadequate touch detection would force users to adapt their control and slow them down. The required accuracy is probably in the *sub-millimeter* range, as we report in Section 3.3 that a 1 mm variability in height measurements resulted in some mis-detections and false activations.

While touch detection appears to be adequate with capacitive surfaces, they do not provide hover tracking. Extending capacitive sensing above the surface was experimented by Hinckley et al. using self-capacitance (Hinckley et al., 2016). However, the approach suffered from low spatial resolution and very perceivable latency. Keeping the standard capacitive sensing of a smartphone for touch detection, Matulic et al. combined it with a camera and used computer vision to track hovering fingers (Matulic et al., 2021). The system's end-to-end latency was measured at 175 ms; which is too high for users' performance estimations. Instead of using a camera, Choi et al. equipped a small surface with LEDs and photo sensors for 2D tracking in addition to

an electrode for capacitive touch sensing (Choi et al., 2011). Even though a video of the system shows good accuracy and low latency, the authors did not report on the accuracy of the approach. In addition, the hover range was limited to 10 mm; which may not be enough for large amplitude target acquisitions. More critically, these three studies were prototyped on small form factors (e.g. a smartphone), but capacitive sensing does not scale well to large surfaces: it is unclear if any of these approaches could scale to a large desktop.

On large surfaces, finger hover and touch tracking has been implemented using alternatives to capacitive sensing, but at the cost of reduced accuracy of touch detection. Harrison et al. used a structured light depth camera and reported touch accuracy in the range on 10 mm (Harrison et al., 2011). With a stereo vision setup, Malik and Laszlo used the camera disparity to estimate the contact of fingers on the surface and reported that "holding the finger approximately 1 cm above the surface works well" (Malik and Laszlo, 2004). Kim et al. improved on the precision of the stereo vision approach (Kim et al., 2014) and reported a vertical range of 7.5 mm "to capture 95% of the touches". Overall, these efforts achieved a touch detection accuracy in the order of 10 mm; which is an order of magnitude larger than the sub-millimeter requirement. Agarwal et al. specifically attempted to implement high accuracy touch detection with the stereo vision approach (Agarwal et al., 2007). They improved on previous work approached by fitting an ellipsoid to the fingertip and implemented touch detection by using a threshold on the altitude of the center of the ellipsoid. They reported "a precision of 2–3 mm", a notable improvement from previous work, but still well above the sub-millimeter requirement.

Vision and stereo-vision approaches have the benefit of operating on bare fingers, whereas marker-based infrared tracking systems offer off-the-shelf sub-millimeter precision at the cost of attaching markers to the fingers. Using a single marker attached on the top of the finger is a common way to get an estimate of the finger position in 3D space (Joshi and Vogel, 2019; Louis et al., 2020). However, the tracking system only reports on the position of the marker, not the exact position of the contact of the pulp of the fingertip on the surface. Attaching a marker at this location is not an option, as the finger would occlude it. The offset between the contact location and the marker on top of the finger can be calibrated by asking participants to touch known locations on the surface. However, this offset depends on the finger orientation when contacting the surface; which cannot be retrieved from a single marker. Using this offset calibration strategy, Joshi and Vogel used a touch threshold at 3 mm and a release threshold at 10 mm (a 7 mm precision). When a rigid-body with three or more markers is attached to the finger, the optical tracking system can recover the 6 degrees of freedom (dof) of the finger; which could be used to improve the estimation of the touch location. Xia et al. attached a 3-markers rigid body to the index finger of participants to predict finger trajectories (Xia et al., 2014). However they reported that the finger "was tracked in 3D" and they did not mention how the touch location was derived from this information. Yeh et al. also attached a 3 markers rigid body to a finger (Yeh et al., 2020). While they did not report on how the tracking information was used, we learned from direct exchanges with the authors that only the 3D position was used to estimate the position of the finger along a device's edge: touch was detected electrically with copper bands.

In summary, many approaches to tracking fingers above the surface and detecting touch events were explored in previous work, but none of them appears to be adequate to the requirement of our experiment: operating on a desktop, i.e. a surface notably larger than that of mobile devices, and with good enough performances, particularly in terms of touch detection, that the system does not hinder participants' performance.





Fig. 1. The experimental setup. A participant is sitting at the desk, pointing with the index finger on the surface of the desk, and looking at the pointer and targets on the display. Optical tracking cameras are setup around the workspace. A small rigid body is attached to the index finger.

### 3. Modeling the fingertip as a sphere

#### 3.1. Principles

Our approach uses marker-based optical tracking as it provides sub-millimeter tracking information. Contrary to previous approaches that only used 3 dof data, even when using multiple markers, we use the full 6 dof of the tracking data. We estimate a contact point position in this coordinate system. In order to account for the different contact location on the pulp of the fingertip depending on the finger's orientation, we model the tip of the finger as a sphere. The finger 2D contact position on the surface, both when hovering and when touching, can be simply computed by projecting the sphere's center on a plane as illustrated in Fig. 2. Touch and release detection is implemented using two thresholds on the altitude of the sphere's center. We use a small hysteresis to prevent touch/release instabilities.

Contrary to Xia et al. (2014), we attached the rigid body to the distal phalanx rather than the intermediate phalanx to ensure that there was no flexible joint between the rigid body and the fingertip. Agarwal et al. modeled the fingertip as an ellipsoid rather than a sphere (Agarwal et al., 2007). An ellipsoid adds two dof compared to a sphere (i.e. 3 principal axes vs. 1 radius) which may allow a more faithful modeling of the bottom surface of the fingertip, but also increases the risk of over fitting; which increases the risk of tracking instabilities. In addition, Agarwal et al. detected contact with a simple altitude threshold criterion on the center of the ellipsoid, similar to our approach. In doing so, they did not use the additional dof of the ellipsoid model and reverted to a spherical criterion for touch detection. As detailed below, we found that modeling the fingertip by a sphere was adequate to detect touch and release events with high accuracy, hence we did not test higher dof models.

Calibrating the sphere model amounts to estimating a value for the 4 dof of the sphere (3 dof for the position in the rigid body coordinate system, and a radius). First, we computed the plane equation  $p$  of the surface by recording the positions of a single marker sliding on the surface and by doing a least square regression analysis. Then, each time the rigid body was attached to a participant's finger, we recorded a sequence of finger motion where the participant always kept a light

contact of the fingertip to the surface but varied the finger position and orientation. This resulted in a recording  $R$  of many data frames, each one providing the 6 dof  $r$  of the rigid body in the world's coordinate system. In each of these frames, we assumed that the distance of the sphere's surface to the touch surface's oriented plane should be zero, as the finger maintained a light touch. We thus searched for the sphere position and radius that minimizes this distance. In other words, to optimize the set of 4 parameters  $s$  (the sphere's dof) over  $R$ , we minimized the error  $E(R, s)$  computed as the sum of sphere-surface distances in all the frames:

$$E(R, s) = \sum_{r \in R} |p(r(s_{position})) - s_{radius}| \quad (1)$$

where  $s_{position}$  is the 3 dof position of the sphere (expressed in the rigid body coordinate system),  $s_{radius}$  its radius,  $r(s_{position})$  transforms the coordinates of the sphere's center from rigid body coordinates to world coordinates, and  $p(x)$  is the signed distance between a world point  $x$  and the surface's plane. We used a gradient descent algorithm for the minimization.

#### 3.2. Hardware setup

We used a 10 cameras Optitrack Prime 13 optical tracking system running at 240 Hz. The Optitrack software ran on a dedicated computer that sent marker positions to the main computer over the network. We used a 2560 × 1440 pixels gaming monitor, with a 597 × 336 mm screen, low input lag (around 10 ms), and a 165 Hz refresh rate. The custom-made C++ software was optimized for low latency. The monitor was set on a desk that was also used as the touch surface, as illustrated in Fig. 1. We calibrated the desk's plane as explained above, and we defined its origin so that the touch area was aligned in front of the display. No visual indication of the touch area on the desk was necessary as users had their visual focus on the monitor: they saw when the cursor was moving out of bounds. As the distal phalanx is the smallest one, the rigid bodies that we printed had to be small: they fitted in a sphere of around 25 mm of diameter and were carrying three 3 mm markers. Their low weight (less than 2 g) allowed us to firmly attach them with double-sided tape as illustrated in Fig. 2.

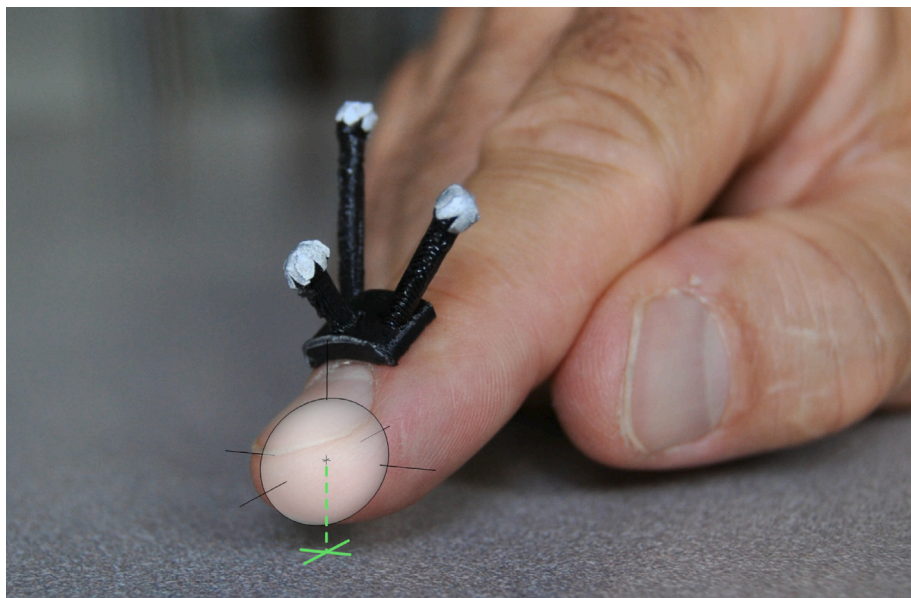


Fig. 2. A 3-marker rigid body attached to the distal phalanx of the index finger. The sphere and lines were added to illustrate the modeling of the fingertip.

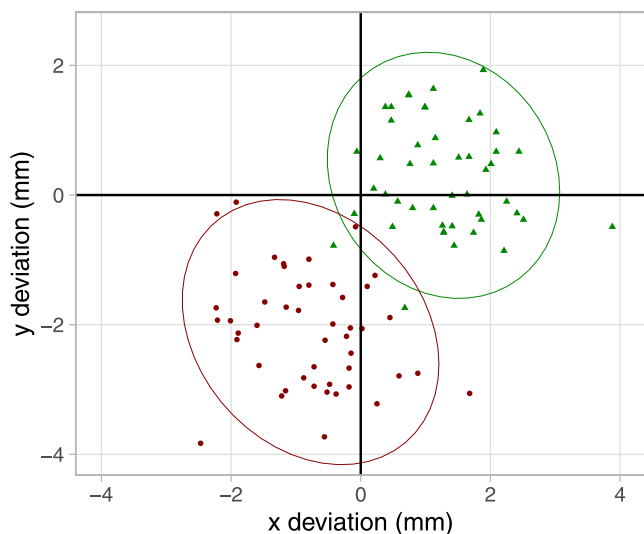


Fig. 3. Precision and accuracy of the system. Deviation of touch locations for the 48 displayed targets. Measured by the iPad capacitive surface (green triangles) and our system (red dots), with 95% confidence ellipses.

### 3.3. Evaluation

We measured the end-to-end latency of the system using the predictive approach from Cattani, Rochet-Capellan and Bérard (Cattani et al., 2015): eight operators measured latencies in the range 25.0–26.0 ms using 0.5 ms steps. The average was 25.7 ms and the standard deviation was 0.3 ms.

To calibrate the sphere, we recorded motion sequences of at least 10 s, providing at least 2400 frame of rigid body 6 dof positional data. The gradient descent took around one second to compute. We checked the accuracy of the process by computing the average error per frame:  $E(R, s)/n(R)$  where  $E(R, s)$  is defined in Eq. (1) and  $n(R)$  is the number of frames in  $R$ . We typically observed values below 0.6 mm. This sub-millimeter average error provided a first indication of the precision of our approach.

We measured the static precision of our approach. We attached the rigid body on an inert object and recorded 9 sequences of around 5 s

of tracking (more than 1200 frames of data per sequence). The object was moved to a new location between recordings. The variability of the measures for a static object was estimated as follows. In each recording, we computed the center of the projections of the sphere on the desk, and the average altitude. Then, we computed the position and altitude deviations by computing the average distance of each observation to these centers. On the nine recordings, the position deviations were in the range 0.043–0.061 mm and the altitude deviation were in the range 0.0034–0.024 mm. Using a control-display gain of 1, one pixel of the display corresponded to a translation of 0.23 mm on the desk. As a result, the system produced a pixel-stable output. This was confirmed by the observation of a stable pointer on the display.

To compare the 2D accuracy of our approach to that of capacitive sensing, we attached markers on a 12.9 inch iPad Pro and calibrated our system's coordinates to match those of the iPad. We then recorded the output of both the iPad capacitive surface and our system when pointing to 12 large crosses regularly spaced on the display, with 4 repetitions. We regrouped the data from the 48 contacts by removing the coordinates of the targets. The result is shown in Fig. 3. We computed the two centers of touch locations for our system and the iPad. Our system center was at 2.27 mm from the target center, 73% more than the iPad's center (1.31 mm). The average distance of our system's touch locations to their center was 1.15 mm; which was 9% larger than for the iPad (1.05 mm). The precision was thus close to that of capacitive sensing. The measured accuracy was not as good, but a notable part of this error may have come from the calibration of the transformation for the tracker's to the iPad's coordinate systems. In any cases, the accuracy of the 2D position on the display is less of a concern in CIT interaction than in direct touch as user's gestures are controlled by the pointer position on the display.

The accuracy of altitude measurement was more critical to our approach as it was used for triggering touch and release events. After calibrating participants' sphere parameters, we asked them to test the touch detection. We observed that the ideal threshold on the sphere's altitude was not at 0 mm, but at a higher value. This can be explained by the difficulty to maintain the lightest touch while recording the calibration sequence: it is easier to press a little more in order to avoid losing contact with the surface. As a result, the sphere is calibrated too low in altitude, and the touch threshold needs to be raised. For each participant, we interactively modified the threshold using steps of 0.1 mm until the participant identified the best threshold. This added

around two minutes to the calibration duration. Typical threshold values ranged from 1 mm to 3 mm. As for the hysteresis, we found that 0.2mm worked well for all participants: we did not observe any touch/release instability, and the release delay induced by the hysteresis remained unnoticed by the participants.

We observed that a trade-off had to be made on the threshold value depending on the *location* of the contact on the surface: activation could happen a few tenth of a millimeter *before* contact in some areas of the desk, and could require more than the slightest contact in other areas. The range of the ideal threshold depending on location was around 1 mm at worst and did not seem to hinder the activation. We considered that this would not be a problem for the experiment. We discuss later that this actually may have been the source of some false or failed activations.

By sliding a marker on the whole surface of the desk, we realized *after the experiment* that altitudes ranged from  $-1.3$  mm to  $0.7$  mm. In other words, the curvature of the desk required a more accurate model than a plane. The problem was mitigated in our experiment, as we only used a small central area of the desk that had limited altitude variation. After the experiment, we verified that partitioning the active area of the desk into 2 cm squares and recording an altitude correction for each of them eliminated the spatial variability of the ideal threshold.

In summary, our approach affords stable and accurate finger hover tracking and touch detection on the large surface of a desktop with low latency. The most critical improvement is touch detection, as we could calibrate a touch threshold with a 0.2 mm precision regardless of finger location and orientation. This is an improvement of an order of magnitude compared to values reported in the literature (2–3 mm (Agarwal et al., 2007), 7.5 mm (Kim et al., 2014), 7 mm (Joshi and Vogel, 2019)).

#### 4. Congruent indirect touch vs. mouse pointing performance

##### 4.1. Goal and hypotheses

The main objective of our study was to question if the superiority of direct touch pointing performance over the mouse, as observed in several previous studies, applied to CIT. If this was the case, we also investigated if this held past the 6 mm size threshold were direct touch performance drops.

CIT being almost nonexistent in common usage, we anticipated that previous experience of our participants would be heavily biased towards the mouse. We opted to measure *learning curves* over several days. We hypothesized that the learning curve would be steeper with CIT than with the mouse, as the learning would only relate to the task in the mouse condition, while it would also relate to the interaction in the CIT condition. We were particularly interested to assess the performance differences in the first and last sessions. We anticipated that mouse performance would be higher than CIT in the first session due to the prior experience of participants. We did not know what to expect for the last session: CIT had never been compared to the mouse, it was unclear how much proprioception could help in the pointing performance of CIT (Gilliot et al., 2014), and we were not sure if the performance advantage of direct touch could transfer to CIT.

##### 4.2. Task and conditions

In a longitudinal study, the choice of tasks and conditions must be specially selective because the duration of each experimental session is highly constrained: it is only a fraction of the participant's total availability for the experiment. Choosing a task for a comparison of interaction techniques is delicate: various tasks may be more efficiently performed by different techniques. For example, input on a digital numerical keyboard may be more efficient with CIT than with a mouse because the constant gain is more suitable than the dynamic gain for the proprioceptive learning of the targets. In contrast, distant targets on very large displays may benefit more from the dynamic gain of the

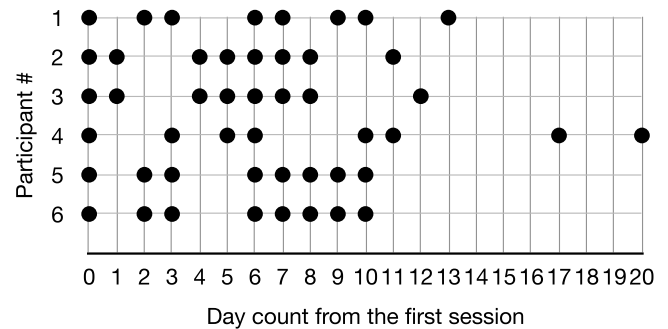


Fig. 4. Participant's schedule of sessions. Each dot represents a session.

mouse compared to the gain 1.0 of CIT. Besides, various techniques have different usage contexts. For example, CIT is expected to be used in a multitouch interaction, where the non-dominant hand can define a spatial reference for the dominant hand (Guiard, 1987), providing for efficient pointing on very large displays (Malik et al., 2005). As a first effort to study the viability of CIT on the desktop, we reverted to the common choice of following the ISO9241-9 standard (Soukoreff and MacKenzie, 2004). This standard for evaluating computer pointing devices relies on target acquisitions of various difficulties; which can be seen as a fundamental building block to build more complex interactions.

When selecting the experimental conditions, we aimed to compare CIT performance with a baseline on the desktop. Mouse and touchpad interactions were the two obvious candidates. However, the mouse has been consistently measured as being notably more efficient than the touchpad (Faizan et al., 2021; Kar et al., 2015; Malekar et al., 2016; Soukoreff and MacKenzie, 2004). One of the reason may be that touchpads do not provide hover tracking: they rely on the fingers sliding in contact with the surface. Sliding fingers on the surface was found to reduce pointing performance in direct-touch (Cockburn et al., 2012; Forlines et al., 2007). We compared CIT with the mouse, assuming that if we observed a performance benefit of CIT over the mouse, the benefit would be even greater when compared to the touchpad. However, while previous studies indicate that this should be true at a *population* level, individual differences may be observed for people that use touchpads exclusively.

##### 4.3. Experiment design

We designed a comparative within-subject Fitts experiment. The two experimental conditions were MOUSE and CIT. A pointer with the shape of a white cross was always visible. In CIT, the finger moved *above* the surface towards the target and landed on the target. The first contact with the surface was used as the validated position. The pointer was shown in green when the finger was in contact with the surface. While more advanced representations of the hand and fingers could help in a real-world use of an CIT desktop (Efanov and Lanir, 2015), we chose to keep a simple cross for consistency with MOUSE and because the risk of confusion in our single finger task was limited.

In order to observe learning curves, we designed a longitudinal study where we favored the number of sessions per participant rather the number of participants. Six participants executed eight sessions on several days, with no more than one session per day. We aimed at no more than three days between sessions, but this was not always possible due to participants' constraints. The full session schedule is provided in Fig. 4. The length of a participant's daily session was calibrated to around half an hour in order to limit the fatigue effect and to facilitate the implementation of the longitudinal study. To balance the order of presentation of conditions, half of the participants began with MOUSE in their first session, the over half began with CIT. Then, at each



**Table 1**

Fitts' IDs (as computed in Soukoreff and MacKenzie (2004)), target widths and amplitudes.

| ID (bit) | Width (mm) | Amplitude (mm) |
|----------|------------|----------------|
| 2        | 12         | 36             |
| 3        | 9          | 63             |
| 4        | 6          | 90             |
| 5        | 4          | 124            |
| 6        | 3          | 189            |

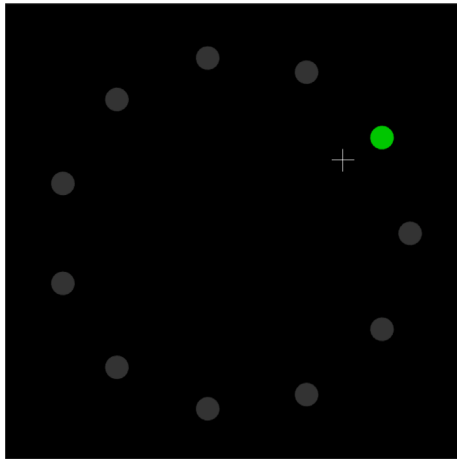


Fig. 5. The target acquisition task for Fitts' ID = 5 ( $w = 4$  mm,  $a = 124$  mm). The cross-shaped pointer is moving towards the next target that is shown in green as the previous acquisition was successful.

new session of a participant, the order was switched from the previous session.

The task was a circular reciprocal pointing task as recommended by Soukoreff and MacKenzie (2004). We used targets with five different Fitts' index of difficulty (ID) in the range 2–6. To reduce participants total experiment time, we used a single couple { width, amplitude } per ID, as detailed in Table 1. For each ID, participants acquired 11 targets represented as small discs regularly spaced on a circle with a radius set at the ID's amplitude. They performed 4 turns of the circle per ID, or 44 targets. The order of presentation of IDs was fixed: 6, 2, 5, 3, 4, and such a block of 5 IDs was repeated twice per session and condition. At the beginning of a new condition, participants warmed up with 44 targets of ID 6 and 44 targets of ID 2. Warmup occurred at every session.

When a new circle of targets corresponding to a new ID was presented, we told participants that they could take a break as the acquisition of the first target was not timed. Otherwise, we told them to go as fast as possible but to adapt their speed so that they missed roughly 2 targets per ID; which is the closest to a 4% error rate on 44 targets. We showed participants that they missed a target by displaying the next target in red, otherwise it was displayed in green as illustrated in Fig. 5.

We used libpointing (Casiez and Roussel, 2011) to implement a reproducible transfer function in the MOUSE condition. The sensitivity was set to 0.875 by default; which corresponds to the default sensitivity on macOS. On the first day, we asked participants to experiment with the mouse and chose the sensitivity that they were the most comfortable with. This chosen sensitivity was then used in all sessions of the participant and is detailed in Table 2. We let participants select their preferred sensitivity, knowing that various participants use different sensitivities in their daily use of a mouse. We assumed a participant would be the most efficient in MOUSE for a sensitivity that matched the one they heavily trained with.

We recorded the target acquisition time and the pointer position at validation. At the end of each turn, we computed the throughput ( $TP$ ,

in bit/s) achieved by the participant on the 11 targets. We used the bivariate model and the mean-of-means computation method for the throughput, as recommended by Wobbrock et al. (2011).

At the end of each session, we recorded participants' general comments on the session. At the end of the last session, we asked participants which of the two interaction techniques they would choose if they had to do a one-hour pointing session on the next day.

#### 4.4. Participants

We recruited 6 volunteer participants from our lab, 3 females and 3 males, aged in the range 18–43 (average 26.3). We refer to them as P $x$  where  $x$  is in the range 1–6. We were able to form a group of participants with diverse practices of desktop and mobile computing, as indicated in Table 2. Of particular interest, we note:

- P1 didn't use a mouse on a regular basis and should be considered as "heavily biased towards relative pointing" rather than "towards the mouse" (one of our hypotheses),
- P3 reported playing first-person shooter games on a daily basis; which provided intense training for mouse pointing,
- P5 regularly performed graphical design using a stylus on a digitizing tablet, i.e., the participant practiced Similar Indirect Pointing on a regular basis, although with a stylus on a small tablet (a control-display gain greater than 1).
- P6 spent significantly more time on a smartphone than on a desktop computer, and was thus more experienced in direct touch than in indirect interactions.

#### 4.5. General observations

##### 4.5.1. Unintended motion on validation

We quickly observed that validation in CIT suffers from an unintended downward translation of the cursor. After hovering on top of the target, the fingertip must land *vertically* on the target to validate the acquisition. However, vertically landing the fingertip when the hand is in contact with the surface is unnatural and difficult to perform. The natural motion is to rotate the joints between the phalanges; which yields a downward motion of the projection of the fingertip on the surface. We observed that two factors controlled the amplitude of this translation: initial fingertip altitude, and the joints involved in the landing. The higher the altitude of the fingertip before validation, the larger the unintended translation. A frequent mitigating strategy of our participants was to maintain the fingertip at a low altitude, e.g., less than 3 mm, during the hover phase. Some participants kept their index finger straight to land it using only the joint between the metacarpus and the proximal phalanx. This increased the radius of the fingertip motion in space, and thus reduced the unintentional pointer motion.

Each participant usually performed very similar validation gestures for all targets. The downward translation was thus almost the same every time. This was sometimes used as a mitigating factor: some participants expressed that they learned the length of the unintended translation and deliberately aimed at a counteracting offset from the target.

##### 4.5.2. Hand poses

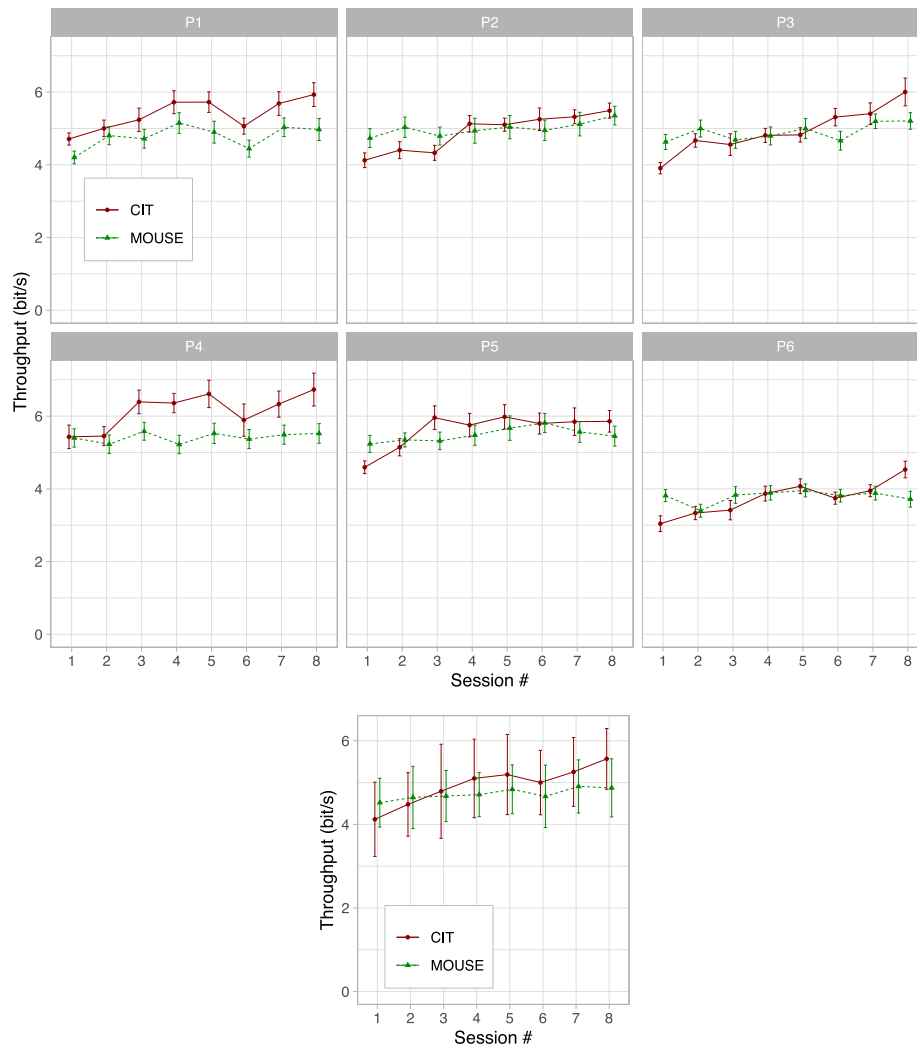
We also observed that participants used various trade-offs when searching to maintain a firm contact of the hand with the surface to stabilize the index finger while trying to limit the friction of the hand when translating on the surface. Participants usually tried several hand poses, sometimes changing the pose between sessions, until they settled for the one that they found the most efficient. We chose *not* to force any specific hand pose in order to observe participants' maximum performance with CIT, regardless of the hand pose. We observed two different ultimate strategies. Three participants (#1, #5, #6) used the "knuckle" pose: they slid the knuckle at the distal joint of the middle,



**Table 2**  
Participants self-reported computing practice, and libpointing sensitivity chosen for the experiment.

| Participant | Mobile/desktop computer use | Mouse/trackpad use on desktop | Chosen mouse sensitivity |
|-------------|-----------------------------|-------------------------------|--------------------------|
| P1          | 50/50                       | 0/100                         | 0.875                    |
| P2          | 50/50                       | 100/0                         | 0.875                    |
| P3          | 70/30                       | 50/50                         | 0.6                      |
| P4          | 5/95                        | 15/85                         | 0.92                     |
| P5          | 30/70                       | 60/10/30 <sup>a</sup>         | 0.875                    |
| P6          | 85/15                       | 100/0                         | 0.875                    |

<sup>a</sup> Mouse/trackpad/stylus on digitizing tablet.



**Fig. 6.** Throughput per session and condition: averages and 95% confidence intervals. Top: for each participant, averaged over each of the 4 turns  $\times$  5 IDs  $\times$  2 repetitions per condition ( $n = 40$ ). Bottom: averaged over the participants ( $n = 6$ ).

ring, and little fingers while these fingers were folded as in a fist. In this pose, the thumb was not touching the surface but rather used to provide stabilization to the index finger, as in Fig. 2. The other three participants used a “spider” hand pose: they slid on the nail tips of all the fingers but the index finger. The hard contact between the desk and the nail tips produced little friction. However, this hand pose may not have been very comfortable as stress in the fingers was sometimes reported.

#### 4.6. Results

We checked that Fitt’s law was a good predictor of pointing time for CIT. We computed linear regressions of the relationship between each participant’s effective ID and pointing time for each session. In

MOUSE, the average R-Squared was 0.99 with a minimum of 0.93. It averaged 0.98 in CIT with a minimum at 0.93, showing a strong fit in both conditions.

##### 4.6.1. Effect of training

The measured throughputs ( $TP$ ) are plotted in Fig. 6. We used the Shapiro–Wilk test to check that the data followed a normal distribution ( $p > 0.025$  in all groups) and Levene’s test to check that variance was homogeneous ( $p = 0.89$ ). We analyzed the data with a two-way repeated measure ANOVA to study the effect of the session and the condition on the throughput. There was no clear effect of the condition ( $F(1,5) = 2.025, p = 0.214$ ), a robust effect of the session on the throughput ( $F(7,35) = 17.4, p = 1.11e-19$ ), and a strong interaction between sessions and conditions ( $F(7,35) = 9.53, p = 1.42e-16$ ).

**Table 3**

Values of the Fitts' experiment parameters measured in the last session, per Index of Difficulty (ID), averaged over the participants. MOU stands for the MOUSE condition.

| ID (bit) | ID <sub>e</sub> (bit) |      | Width <sub>e</sub> (mm) |       | Time (s) |      | Error rate (%) |      | TP (bit/s) |      |
|----------|-----------------------|------|-------------------------|-------|----------|------|----------------|------|------------|------|
|          | CIT                   | MOU  | CIT                     | MOU   | CIT      | MOU  | CIT            | MOU  | CIT        | MOU  |
| 2        | 1.82                  | 1.51 | 14.2                    | 18.9  | 0.27     | 0.37 | 5.12           | 11.4 | 6.94       | 4.17 |
| 3        | 2.42                  | 2.40 | 14.7                    | 15.3  | 0.46     | 0.52 | 13.8           | 9.86 | 5.29       | 4.76 |
| 4        | 3.52                  | 3.48 | 8.86                    | 8.98  | 0.64     | 0.67 | 8.72           | 7.20 | 5.59       | 5.28 |
| 5        | 4.37                  | 4.54 | 6.46                    | 5.65  | 0.80     | 0.84 | 10.4           | 7.39 | 5.59       | 5.52 |
| 6        | 5.23                  | 5.43 | 5.28                    | 4.57  | 1.00     | 1.01 | 15.0           | 9.48 | 5.38       | 5.45 |
| mean     | 3.47                  | 3.47 | 9.89                    | 10.69 | 0.63     | 0.68 | 10.6           | 9.06 | 5.76       | 5.04 |

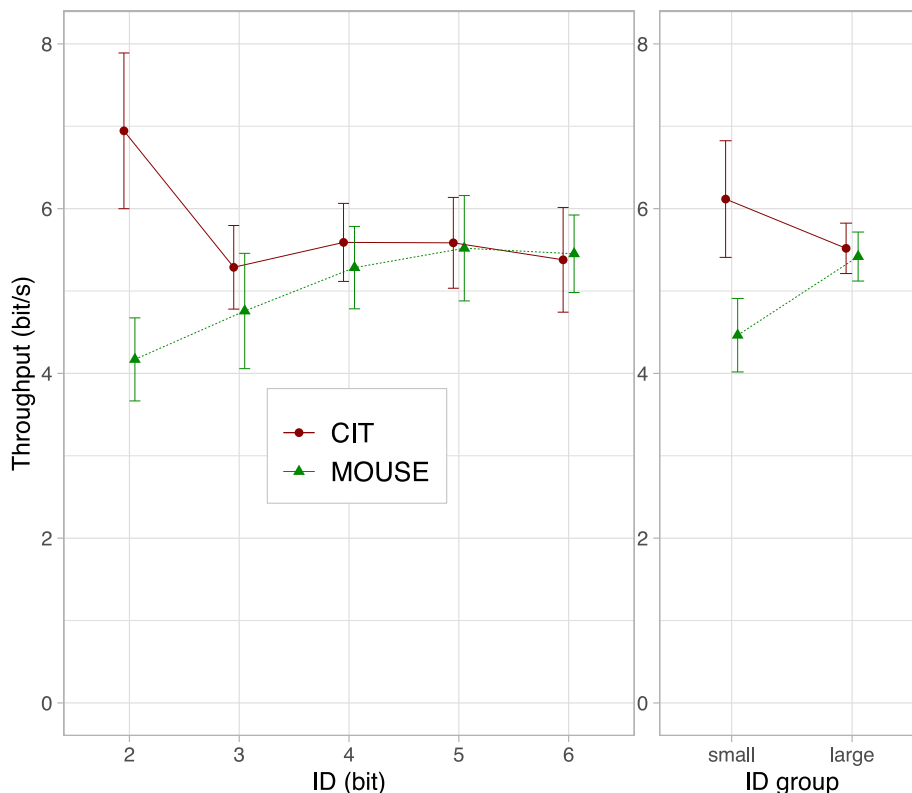


Fig. 7. Throughput per ID on the last session, averaged over the 6 participants, with 95% error bars. Left: for each ID. Right: grouped by small ( $ID \leq 3$ ) and large ( $ID \geq 4$ ) ID.

We averaged the throughput per condition across all participants in the first session and in the last session and tested the differences with paired samples t-tests. For MOUSE, the average throughput increased from 4.52 bit/s to 4.87 bit/s but the effect of learning was rather weak ( $t(5) = -2.82, p = 0.0372$ ). For CIT, the learning effect was very robust ( $t(5) = -11.2, p = 9.95e - 5$ ) with throughput increasing from 4.12 bit/s to 5.57 bit/s.

#### 4.6.2. Performances after training

To compare the performance of CIT and MOUSE *once participants have trained with CIT*, we focus our analysis on the last session. Table 3 summarizes the parameters of the Fitts' experiment. We checked the difference in error rates between conditions. Error rates averaged 9.05% in MOUSE and 10.6% in CIT, revealing that participants were more oriented towards speed than accuracy. The difference was not robust ( $t(5) = 1.49, p = 0.1952$ ). All participants produced a better throughput in CIT than in MOUSE. The ratio ranged from 1.03 for P2 to 1.21 for P6. When averaged over the six participants (Fig. 6, bottom, session 8), CIT throughput was 14% higher than mouse throughput. A paired sample t-test revealed that this effect was robust ( $t(5) = 4.73, p = 0.005175$ ). We estimated the effect size with Cohen's D formula. At 1.029 pooled standard deviation, the effect size was between "large" (0.8) and "very large" (1.2) (Sawilowsky, 2009).

We plotted the throughput per Fitts' ID to inquire into the effect of target difficulty. Fig. 7 reveals that the largest gains in CIT throughput came from the easiest targets. We grouped IDs as "small IDs" ( $ID \leq 3$ ,  $width \geq 9$  mm) and "large IDs" ( $ID \geq 4$ ,  $width \leq 6$  mm). For small IDs, the CIT throughput averaged over the participants was 37% greater than in MOUSE ( $t(5) = 9.08, p = 0.000270$ ). We also observed that error rates were similar for  $ID \leq 4$  ( $t(5) = -0.155, p = 0.883$ ), but they were higher in CIT than in MOUSE for large  $ID \geq 5$  (12.7% vs 8.43%,  $t(5) = 4.73, p = 0.00521$ ).

#### 4.6.3. Participants' comments

As explained in Section 3.3, touch detection was not perfect. Mis-detections were sometimes reported by participants. They reported both false activations when touch was detected as the fingertip was still hovering, and missed activations when touch on a target was not registered by the system. When asked about an estimation of the number of mis-detections, participants provided answers in the range 5 to 10; which roughly corresponded to our observations. Hence, mis-detections occurred in small numbers with respect to the 440 activations of an CIT session. These reports occurred on session 1 for P2 and P5, sessions 2 and 6 for P4, session 1, 6 and 7 for P6. P4 expressed that the effect of a touch mis-detection was probably more severe than simply affecting the corresponding target. The participant would become less confident

about the system and adapt its control to hover a little higher and press a little harder on the desk. This would slow down subsequent target acquisitions.

Four participants commented about the fatigue that they felt during the sessions. P2, using the “spider” hand pose, felt some stress in his fingers at the end of most CIT sessions. CIT sessions also generated fatigue in the shoulder for participants #2 and #6. Participants #3 and #4 felt fatigue in the wrist during the MOUSE sessions: P3 expressed that longer breaks were required in MOUSE than in CIT. P4 expressed that in MOUSE the need for a break was felt before the end of the 44 targets.

While the motion strategy was constant in MOUSE; all participant commented that they used different strategies in CIT depending on the target difficulties. In  $ID = 2$  in particular, the target width was large enough that some participants did not need to stabilize the hand with a contact with the desk: the whole hand was hovering above the surface. This saved time by removing the need to slide the hand. The small amplitude at this ID allowed other participants to keep their hand or their wrist anchored to the desk as moving the index finger was enough to reach all targets of the circle. Here again, sliding the hand was unnecessary. Some participants commented that moving their visual focus on the next target was too slow. They chose to keep it at the center of the circle of targets, using only their peripheral vision to detect failures with red targets. Thanks to these strategies, participants expressed that they felt very efficient in CIT for  $ID = 2$  targets. Some participants reported their attempt to reproduce some of these strategies at  $ID = 3$  but with mixed results.

P4 expressed several appreciative comments on CIT. It felt “magical to validate on the display by tapping on a non-instrumented desk”. When we noted that the finger was instrumented, the participant had forgotten about the rigid body due to its negligible weight. CIT also gave the feeling of “holding a mouse with no weight and no friction” due to the use of the “spider” hand pose. In particular, the static friction that must be overcome to initiate motion was found to be lower with the nails than with the mouse. The lack of *drift* was also appreciated: the mouse tended to lose its ideal location on the desk during a run but the participant did not want to lose time by clutching. Finally, the participant felt that it was easier to learn the position of the targets in CIT than in MOUSE. This allowed to initiate more powerful ballistic motions that ended closer to the target.

We asked participants after the last session which of the two interaction techniques they would choose if they had to do a one-hour pointing session on the next day. Participants #1, #2, and #3 chose the mouse. This was justified by: the mouse always working the same whereas touch requiring to adapt the hand pose depending on the target difficulty (#1), the feeling of being more efficient and requiring less efforts (#2), and the fear of touch generating too much stress on the finger on a long session even though it felt more accurate (#3). The other three participants expressed a preference for CIT. P4 moderated the preference by expressing fear of unintended activation in a real-world desktop context. P5 and P6 linked their preference for CIT to their personal computing practices: the frequent use of a digitizing tablet for P5 and highly dominant use of direct touch for P6.

## 5. Discussion

### 5.1. CIT implementation

The hindrance of our system on users’ pointing performance was limited as indicated by participants being able to equal or outperform the mouse, a highly optimized interaction that benefited from decades of training for some of our participants. In pilot tests, a user expressed that the system disappeared in his mind and that he felt that the pointer on the screen was his finger. P4 forgot the rigid body that was attached to the finger and appreciated the “non-instrumented interaction”, an

indication that the measures obtained with our system should generalize to non-instrumented solutions. Participants of the study made almost no comment about the technical performances of the system. The only problems came from the precision of the altitude tracking: mis-activation were sometimes reported during the experiment. We identified that this stemmed from the curvature of the desk, and we propose in Section 3.3 that partitioning the surface and measuring an altitude correction in each cell solves the problem.

While our approach only applies to *lab studies* rather than *deployment in the field*, it could be reproduced in other interaction contexts to allow un-hindered touch performance measurements. These include for example direct touch for back-of-device interaction (Gugenheimer et al., 2016; Wigdor et al., 2007) and shape-changing interfaces (Alexander et al., 2018). Putting aside the 10-camera optical tracking system, the fingertip modeling as a sphere and the calibration using the finger contact on a known surface should be suitable to more deployable approaches, such as stereo vision (Agarwal et al., 2007; Kim et al., 2014).

### 5.2. CIT vs. mouse pointing performance

This study provides first measurements of CIT performances in a desktop context. Five out of six participants realized higher throughput in MOUSE than in CIT in the first session, one participant having similar throughput in both conditions. The last session revealed a reverse situation, underscoring the importance of longitudinal studies. After 8 training session, the pointing throughput of CIT was similar to that of the mouse for participants #2 and #5, and was greater for the four other participants. Overall, the performance improvement is notable: more than 19% for participants #1, #4 and #6, and 14% when averaged over all participants. These are first evidences that CIT may be a more efficient pointing interaction than mouse on the desktop.

In their review of Fitts’ law studies conforming to ISO9241-9, Soukoreff and MacKenzie observed that the range of throughput values for the mouse was 3.7–4.9 bit/s (Soukoreff and MacKenzie, 2004). As most measures were performed in a single session, we compare this range to the one in the first mouse session of our participants. At 3.8–5.4 bit/s (mean = 4.7 bit/s), our participants had performances that were coherent with the ones observed in previous work. In other words, their mouse pointing abilities were not out of the ordinary. This is a promising indicator of the generality of these results, although further studies increasing the number of observations will be required for confirmation.

One of our main objectives was to test if CIT would be unaffected by the fat finger problem. We observed a performance drop in CIT from target sizes 12 mm to 9 mm and below, but Fig. 7 indicates that the throughput remained at or above mouse levels. However, high throughput can be achieved by sacrificing accuracy for faster motion. Indeed, although error rates were similar in both conditions when averaged across all targets, results indicate that for the smallest targets (width  $\leq 4$  mm) participants realized larger effective width in CIT than in MOUSE; which resulted in higher error rates. It is unclear if participants could improve their error rate in CIT by slowing down their motion, but this observation suggests that 4 mm may be close to the limit of target sizes that can be efficiently acquired with CIT. In summary, this study provides first evidences that CIT is less affected by the fat finger problem than direct touch, but small targets with sizes of 4 mm or less may be more effectively acquired with a mouse.

The other main objective was to assess if the higher throughput of direct touch compared to the mouse would be observed with CIT. When looking only at the easiest targets with  $ID = 2$  bit, throughput improved by 37% from mouse to CIT. The average throughput achieved across the six participants was 6.94 bit/s; which is remarkably close to the 6.95 bit/s measured by MacKenzie on direct touch targets of IDs in the range 1.14–3.17 bit (MacKenzie, 2015). This is notably higher than maximum throughput per ID achieved by our participants with

the mouse with an average at 5.52 bit/s for ID = 5 bit. In other words, we observed a performance improvement compared to the mouse that was similar to the improvements observed in the literature with direct touch, but only for the targets with the lowest ID. With higher IDs, we only observed a small improvements or similar performances.

Comments from our participants indicate that their control with ID = 2 bit was different from the control with the other IDs. In particular, keeping the visual focus at the center of the target circle, instead of moving it between targets, is an atypical behavior. This may be explained by the learning of the proprioceptive location of the targets, allowing for a faster sensorimotor loop than the visuomotor loop. This did not seem to be possible with the mouse as throughput tended to decrease with lower IDs. It should be noted that rapid target acquisitions with such low IDs are common in direct touch interfaces. This occurs for example when selecting photo vignettes from a photo gallery, or when typing numbers on a numerical keypad. On a mouse desktop, “typing” on a virtual numerical keypad can be a little frustrating, as one may feel that typing with a finger would be notably more efficient. This may explain why such low ID targets are less common on the desktop than on mobile touch devices. Our study indicates that a CIT desktop would be efficient for both large and small targets, i.e. benefiting from the strengths of both direct-touch and desktop pointing.

We defined target IDs by single pairs of width and amplitude in order to keep participants’ experimental time to acceptable levels. In future work, we want to assess the relative contribution of the width and the amplitude of targets in the efficiency of CIT. This will be of particular interest to analyze the throughput spike for ID = 2 bit.

### 5.3. Limitations of the study

We designed our experiment as a longitudinal study knowing that CIT was an unknown interaction, and that participant would require training before achieving relevant pointing performances. The curves of throughput across sessions (Fig. 6) show that, for most participants, the CIT curve is still steeper than the mouse curve on the last session. In other words, participants would probably improve relatively more in CIT than in mouse given more experimental sessions. As a first indication of a longer learning curve, and considering a participant that is heavily biased towards the maximal effort, the author of this paper performed 14 experimental sessions. The best throughput with CIT were recorded on sessions 11 (7.14 bit/s) and 12 (7.15 bit/s). Throughput then declined to 6.93 bit/s and 6.64 bit/s. However, longer longitudinal studies with unbiased participants will be required to get a full picture of the learning curve of CIT.

We provided our best effort to implement a CIT interaction which technical performance was good enough that it would not hinder users’ pointing performances. We observed some failure of touch detection. While each touch detection failure increased the acquisition time of the corresponding target, their effect on the overall measure of throughput was probably more severe as expressed by P4. Overall, these two limitations of our study indicate that the measured CIT throughput should be seen as a *lower bound*: the performance advantage of CIT over the mouse is probably higher.

Our study indicates that CIT has superior performance than the mouse for target acquisition. Yet, desktop interaction is based on other frequent interactions, such as dragging and dropping items and text input with a keyboard. In both cases, difficulties emerge in the context of CIT. We suggest some potential approaches in the next paragraph, but until such solution are tested and are proved efficient, the general efficiency of CIT on the desktop remains an open question.

In this study, we observed a coherent, large, and robust effect on participants having various backgrounds in pointing interaction. Still, as in any controlled study, our experiment tested a very small population compared to the target population. To test the generality of the results, more studies on this seldom investigated form of interaction are required.

### 5.4. On an indirect multitouch desktop

This study provides first evidence of CIT’s high pointing performances, even on small targets. While our study focused on single finger target acquisition, more fingers could be recruited to perform other kinds of multi-touch interactions (Kin et al., 2009; Malik et al., 2005). This raises the question of the viability of an “indirect multitouch desktop” where touching an empty desk would replace the mouse for pointing.

While our implementation is not suitable for deployment, a more compact computer vision system could provide a viable approach. Two cameras embedded in the top corners of the computer’s display could feed a stereo vision system and track the hands, the fingers, and other useful interactive objects such as a pen (Kim et al., 2014).

Pointing performance is an important dimension of the usability of an interaction, but other dimensions have to be considered. Fatigue in the arm and finger stress could be serious issues and should be scrutinized. The requirement for a large empty space on the desk raises the problem of text input: a physical keyboard may remain on the far side of the touch area, close to the display, but at the cost of requiring more desk space and a suboptimal arm position when typing. It may be more suitable to switch to some form of soft keyboard. While text input performance may be lower than on a physical keyboard, soft keyboards have other benefits, such as fast cursor control with gestures (Kim et al., 2013) or fast learning of hotkeys (Fennedy et al., 2022). To reduce the space used for pointing, Similar Indirect Touch with control-display gains greater than 1 should be investigated. This could help both with a fatigue issue and for freeing desk space for a physical keyboard, but at the cost precision. As for dragging interactions, sliding a finger in contact with a touch surface is inefficient on the glass surface of most direct touch devices, but other materials may be more suitable. Alternatively, other gestures could be investigated for dragging digital objects. For example, touching with a near-horizontal finger could attach the digital object to the hovering finger, allowing for drag and drop without contact. Beyond improved performances, it would be interesting to assess if indirectly controlling digital objects with the fingers generates the same feeling of naturalness that has often been associated with direct touch.

## 6. Conclusion

We studied Congruent Indirect Touch (CIT), a slight variation of direct touch that separates the motor and display spaces. CIT has seldom been studied because of the challenge of implementing high precision hover tracking of fingertips. Using a novel fingertip tracking approach, we could observe that CIT pointing performance is overall superior to the status quo (the mouse). CIT had similar performances to mouse on difficult targets with sizes down to 3 mm, and notably superior performances on easy targets (37%). On average, the performance improvement was 14%. Considering the massive use of pointing in desktop interaction, any performance improvement can have tremendous benefits. In addition, CIT opens the way to multitouch interactions on the desktop. This study provides first evidences of the viability of the CIT interaction on the desktop in terms of pointing performances. Other dimensions must be explored such as long-term use and text input performance.

### CRedit authorship contribution statement

**François Bérard:** Conceptualization, Investigation, Methodology, Software, Writing – original draft, Writing – review & editing.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.



## Data availability

Data will be made available on request.

## Acknowledgments

This work was supported by the CONTINUUM project, France ANR-21-ESRE-0030. The author is grateful to Quentin Roy for his advices.

## Appendix A. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.ijhcs.2024.103261>.

## References

- Agarwal, A., Izadi, S., Chandraker, M., Blake, A., 2007. High precision multi-touch sensing on surfaces using overhead cameras. In: IEEE International Workshop on Horizontal Interactive Human-Computer Systems. TABLETOP, pp. 197–200.
- Ahmad, A., Migniot, C., Dipanda, A., 2019. Hand pose estimation and tracking in real and virtual interaction: A review. *Image Vis. Comput.* 89, 35–49.
- Alexander, J., Roudaut, A., Steimle, J., Hornbæk, K., Bruns Alonso, M., Follmer, S., Merritt, T., 2018. Grand challenges in shape-changing interface research. In: ACM Conference on Human Factors in Computing Systems. CHI '18, Association for Computing Machinery, New York, NY, USA, pp. 1–14.
- Boring, S., Jurmu, M., Butz, A., 2009. Scroll, tilt or move it: Using mobile phones to continuously control pointers on large public displays. In: Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7. OZCHI '09, Association for Computing Machinery, New York, NY, USA, pp. 161–168.
- Casiez, G., Roussel, N., 2011. No more bricolage! methods and tools to characterize, replicate and compare pointing transfer functions. In: ACM Symposium on User Interface Software and Technology. UIST, Association for Computing Machinery, pp. 603–614.
- Cattan, E., Rochet-Capellan, A., Bérard, F., 2015. A predictive approach for an end-to-end touch-latency measurement. In: ACM International Conference on Interactive Tabletops and Surfaces. ITS, ACM, pp. 215–218.
- Cause, M., Alonso, R., Vachon, F., Parise, R., Orliaguet, J.-P., Tremblay, S., Terrier, P., 2014. Testing usability and trainability of indirect touch interaction: perspective for the next generation of air traffic control systems. *Ergonomics* 57 (11), 1616–1627.
- Choi, S., Han, J., Kim, S., Heo, S., Lee, G., 2011. ThickPad: A hover-tracking touchpad for a laptop. In: ACM Symposium on User Interface Software and Technology (UIST) Adjunct Proceedings. UIST '11 Adjunct, Association for Computing Machinery, New York, NY, USA, pp. 15–16.
- Cockburn, A., Ahlström, D., Gutwin, C., 2012. Understanding performance in touch selections: Tap, drag and radial pointing drag with finger, stylus and mouse. *Int. J. Hum. Comput. Stud.* 70 (3), 218–233.
- Efanov, I., Lanir, J., 2015. Augmenting indirect multi-touch interaction with 3D hand contours and skeletons. In: ACM Conference on Human Factors in Computing Systems (CHI) Extended Abstracts. CHI (EA), ACM, pp. 989–994.
- Faizan, M.S., Mian, T., Muzammil, M., 2021. A comparative analysis of a mouse and touchpad based on throughput and locations for a laptop computer. In: Muzammil, M., Khan, A.A., Hasan, F. (Eds.), *Ergonomics for Improved Productivity*. Springer Singapore, Singapore, pp. 219–225.
- Fennedy, K., Srivastava, A., Malacria, S., Perrault, S.T., 2022. Towards a unified and efficient command selection mechanism for touch-based devices using soft keyboard hotkeys. *ACM Trans. Comput.-Hum. Interact.* 29 (1).
- Forlines, C., Wigdor, D., Shen, C., Balakrishnan, R., 2007. Direct-touch vs. mouse input for tabletop displays. In: ACM Conference on Human Factors in Computing Systems. CHI, ACM, pp. 647–656.
- Gilliot, J., Casiez, G., Roussel, N., 2014. Impact of form factors and input conditions on absolute indirect-touch pointing tasks. In: ACM Conference on Human Factors in Computing Systems. CHI, ACM, pp. 723–732.
- Gugenheimer, J., Dobbstein, D., Winkler, C., Haas, G., Rukzio, E., 2016. FaceTouch: Enabling touch interaction in display fixed UIs for mobile virtual reality. In: ACM Symposium on User Interface Software and Technology. Association for Computing Machinery, pp. 49–60.
- Guiard, Y., 1987. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *J. Motor Behav.* 19, 486–517.
- Harrison, C., Benko, H., Wilson, A.D., 2011. OmniTouch: Wearable multitouch interaction everywhere. In: ACM Symposium on User Interface Software and Technology. UIST, ACM.
- Hinckley, K., Heo, S., Pahud, M., Holz, C., Benko, H., Sellen, A., Banks, R., O'Hara, K., Smyth, G., Buxton, W., 2016. Pre-touch sensing for mobile interaction. In: ACM Conference on Human Factors in Computing Systems. CHI '16, Association for Computing Machinery, New York, NY, USA, pp. 2869–2881.
- Holz, C., Baudisch, P., 2010. The generalized perceived input point and how to double touch accuracy by extracting fingerprints. In: ACM Conference on Human Factors in Computing Systems. CHI '10, ACM, New York, NY, USA, pp. 581–590.
- Joshi, N., Vogel, D., 2019. An evaluation of touch input at the edge of a table. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. CHI '19, Association for Computing Machinery, New York, NY, USA, pp. 1–12.
- Kar, G., Vu, A., Nehme, B.J., Hedge, A., 2015. Effects of mouse, trackpad and 3d motion and gesture control on performance, posture, and comfort. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, Vol. 59, No. 1. pp. 327–331.
- Kim, D., Izadi, S., Dostal, J., Rhemann, C., Keskin, C., Zach, C., Shotton, J., Large, T., Bathiche, S., Niefner, M., Butler, D.A., Fanello, S., Pradeep, V., 2014. RetroDepth: 3D silhouette sensing for high-precision input on and above physical surfaces. In: ACM Conference on Human Factors in Computing Systems. CHI, ACM, pp. 1377–1386.
- Kim, S., Son, J., Lee, G., Kim, H., Lee, W., 2013. TapBoard: Making a touch screen keyboard more touchable. In: ACM Conference on Human Factors in Computing Systems. CHI, ACM, pp. 553–562.
- Kin, K., Agrawala, M., DeRose, T., 2009. Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. In: Proceedings of Graphics Interface 2009. GI, Canadian Information Processing Society, pp. 119–124.
- Louis, T., Troccaz, J., Rochet-Capellan, A., Bérard, F., 2020. GyroSuite: General-purpose interactions for handheld perspective corrected displays. In: ACM Symposium on User Interface Software and Technology. UIST, pp. 1248–1260.
- MacKenzie, S.I., 2015. Fitts' throughput and the remarkable case of touch-based target selection. In: Proceedings of the International Conference on Human-Computer Interaction. HCII.
- MacKenzie, I.S., Ware, C., 1993. Lag as a determinant of human performance in interactive systems. In: ACM Conference on Human Factors in Computing Systems. CHI, pp. 488–493.
- Maleckar, A., Kljun, M., Rogelj, P., Pucihar, K., 2016. Evaluation of common input devices for web browsing: Mouse vs touchpad vs touchscreen. In: Proceedings of the Human-Computer Interaction in Information Society. pp. 9–13.
- Malik, S., Laszlo, J., 2004. Visual touchpad: a two-handed gestural input device. In: ACM International Conference on Multimodal Interfaces. ICMI, ACM, pp. 289–296.
- Malik, S., Ranjan, A., Balakrishnan, R., 2005. Interacting with large displays from a distance with vision-tracked multi-finger gestural input. In: ACM Symposium on User Interface Software and Technology. UIST, ACM, pp. 43–52.
- Matulic, F., Ganeshan, A., Fujiwara, H., Vogel, D., 2021. Phonetroller: Visual representations of fingers for precise touch input with mobile phones in VR. In: ACM Conference on Human Factors in Computing Systems. CHI '21, Association for Computing Machinery, New York, NY, USA.
- McCallum, D.C., Irani, P., 2009. ARC-Pad: absolute+relative cursor positioning for large displays with a mobile touchscreen. In: ACM Symposium on User Interface Software and Technology. UIST, ACM, pp. 153–156.
- Sasangohar, F., MacKenzie, I.S., Scott, S.D., 2009. Evaluation of mouse and touch input for a tabletop display using fitts' reciprocal tapping task. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, Vol. 53, No. 12. pp. 839–843.
- Sawilowsky, S.S., 2009. New effect size rules of thumb. *J. Mod. Appl. Stat. Methods* 8 (2), 26.
- Schmidt, D., Block, F., Gellersen, H., 2009. A comparison of direct and indirect multi-touch input for large surfaces. In: IFIP Conference on Human-Computer Interaction. INTERACT, Springer-Verlag, pp. 582–594.
- Sears, A., Shneiderman, B., 1991. High precision touchscreens: design strategies and comparisons with a mouse. *Int. J. Man-Mach. Stud.* 34 (4), 593–613.
- Soukoreff, R.W., MacKenzie, I.S., 2004. Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI. *Int. J. Hum. Comput. Stud.* 61 (6), 751–789.
- Tactiva, 2005. Tactapad demo video.
- Wigdor, D., Forlines, C., Baudisch, P., Barnwell, J., Shen, C., 2007. Lucid touch: a see-through mobile device. In: ACM Symposium on User Interface Software and Technology. UIST, ACM, pp. 269–278.
- Wobbrock, J.O., Shinohara, K., Jansen, A., 2011. The effects of task dimensionality, endpoint deviation, throughput calculation, and experiment design on pointing measures and models. In: ACM Conference on Human Factors in Computing Systems. CHI, ACM, pp. 1639–1648.
- Xia, H., Jota, R., McCanny, B., Yu, Z., Forlines, C., Singh, K., Wigdor, D., 2014. Zero-latency tapping: Using hover information to predict touch locations and eliminate touchdown latency. In: ACM Symposium on User Interface Software and Technology. UIST, ACM, pp. 205–214.
- Yeh, Y.T., Roy, Q., Irudayaraj, A.A.R., Vogel, D., 2020. Expanding side touch input on mobile phones: Finger reachability and two-dimensional taps and flicks using the index and thumb. *Proc. ACM Hum.-Comput. Interact.* 4 (ISS).