Reducing Latency with a Continuous Prediction: Effects on Users' Performance in Direct-Touch Target Acquisitions

Elie Cattan[1], **Amélie Rochet-Capellan**[2], **Pascal Perrier**[2], **François Bérard**[1]

LIG, University of Grenoble[1]
Grenoble, France
{first.last}@imag.fr

GIPSA-lab, University of Grenoble[2]
Grenoble, France
{first.last}@gipsa-lab.grenoble-inp.fr

ABSTRACT

Latency in direct-touch systems creates a spatial gap between the finger and the digital object when dragging. This breaks the illusion of presence, and has a negative effect on users' performances in common tasks such as target acquisitions. Latency can be reduced with faster hardware, but reaching imperceptible levels of latency with a hardware-only approach is a difficult challenge and an energy inefficient solution.

We studied the use of a continuous prediction of the touch location as an alternative to the hardware only approach to reduce the latency gap. We implemented a low latency touch surface and experimented with a constant speed linear prediction with various system latencies in the range [25ms-75ms]. We ran a user experiment to objectively assess the benefits of the prediction on users' performances in target acquisition tasks. Our study reveals that the prediction length is strongly constrained by the nature of target acquisition tasks, but that the approach can be successfully applied to counteract a large part of the negative effect of latency on users' performances.

ACM Classification Keywords

H.5.2 User Interfaces: Input devices and strategies (e.g., mouse, touchscreen)

Author Keywords

latency; direct-touch; dragging task; continuous prediction; target-acquisition; user performances

INTRODUCTION

Direct touch is a key element of the success of touch-sensitive devices, giving the sensation of moving digital objects with a direct contact of a finger. However, the immediacy of direct touch interaction and its "naturalness" comes up against a technical limit: the latency of the system, defined as the delay between a user action and the feedback of this action. With direct touch, latency materializes as a spatial gap between the finger and the virtual object under control. Latency in direct touch systems can be perceived at levels as low as 2ms [15], and it was shown to deteriorate users' target acquisition performances at levels as low as 10ms [7]. However, current commercial touch-sensitive devices still suffer from latencies in the range [50ms-200ms] [15], providing a sub-optimal experience to the users and a decrease in performances. The spatial gap induced by latency increases when the finger is moving faster to cover larger distances. This amplifies the negative effect of latency on large interactive surfaces compared to smaller devices such as tablet computers or smartphones. The omnipresent hindrance induced by latency may conceal the benefits of novel forms of direct touch interactions. The compensation of hardware latency is thus a core challenge for interaction designers on large interactive surfaces.

A radical way to eliminate the negative effects of latency is to develop a touch system that has a small enough latency that it is not perceived by users. Ng et al. built such a system and reached 1ms of latency by using custom developed sensor, processing and display hardware [15]. This *pure hardware* approach was successful in providing an experimental device that allowed detailed studies on the perception and on the effect of latency. However, achieving 1ms of latency came at the cost of strong limitations: the system could not display color, it had a width of only 24cm, and the software had to run on an embedded real-time OS, hence it could not rely on common OSes and their graphics libraries. The display part of the system also appears as a bottleneck: going from the vast majority of current 60Hz display refresh rate to the 1kHz refresh rate of Ng et al. [15] requires a 17x improvement which is a major technical challenge. Besides, augmenting the display refresh rate for the sole purpose of reducing the latency gap appears as a very inefficient use of computing and energy resources. As such, the approach does not appear as a practicable solution to the latency problem in general purpose direct touch systems.

Here, we study the benefits of using a *software prediction*. With prediction, the virtual object under control is not displayed at the most recently sensed location of the finger, but rather at a location predicted based on the finger's past trajectory. Compensating latency with prediction has been well studied in various domains such as robotics and virtual reality. In the HCI literature, prediction has been used to anticipate which target a user is aiming at. A *single* prediction was made well before the end of the acquisition in order to guess the *end-point* of a user controlled trajectory [2, 4, 8, 16]. These

ITS 2015, November 15–18, 2015, Funchal/Madeira, Portugal.

ACM 978-1-4503-3899-8/15/11 ...$15.00.
DOI: http://dx.doi.org/10.1145/2817721.2817736

efforts dealt with "large" prediction errors by requiring final user adjustments. The single prediction approach was also successfully used in direct touch in order to achieve "zero-latency tapping" [20]. In this paper, we tackle a different problem as we use the prediction to *continuously* reduce the gap between the finger and the digital object while dragging. This poses the challenge of predicting a "no-latency" position at every sample. However, predicting the future always come with uncertainty, which is a source of tracking mismatch on its own. A predictive approach is only useful if the benefits of reducing the latency gap surpass the negative effects of the prediction mismatch. While device makers have started to use continuous predictions to reduce touch latency on mobile devices [18], the prediction is limited in length (16ms) and only attempts to partially compensate the device's latency. There is no published work that informs about the amount of latency that can be compensated through a continuous prediction and, more fundamentally, that provides an objective evaluation of the benefits of the prediction on users' performances.

In this work, we studied the total compensation of a device's latency through continuous prediction. We selected users' performances in target acquisitions as an objective measure of the benefits of the prediction. The target acquisitions were performed by sliding a digital object onto a target location, which we chose as the most common direct touch dragging interaction. In order to experiment with low levels of system latency, we created an experimental device with a baseline latency of 25ms. By artificially adding delays to this baseline, we could experiment with higher system latencies at 42ms, 58ms and 75ms. Our main objective is to inform about the levels of latency where a continuous prediction is a viable approach to users' performance improvements.

In the remainder of the paper, we review previous work related to the effect of latency on HCI and to the use of users' motion predictions. We then describe our experimental apparatus, and our software prediction approach. We next report on the user study that evaluates the benefits of this prediction on user's performances. We also provide an indication of user's preferences when comparing the system with and without prediction in target acquisitions and free-movement tasks. We finally discuss the results of the study in regard to the effect of latency on touch interaction and the feasibility of prediction in current systems.

RELATED WORK

Effects of latency

The detrimental effect of latency on HCI has been investigated from different viewpoints: from the subjective annoyance felt by users to the more objective measurement of degradations in users' performance. Studying visual immersion, Meehan et al. compared users' self reported sense of presence and measured the change of heart rate when moving from a calm to a stressful virtual environment, under two conditions of latency: 50ms and 90ms [13]. They found that participants in the low latency condition had a higher sense of presence and a higher change in heart rate. More recently, Ng et al. used the just noticeable difference (JND) protocol to objectively explore users' ability to perceive latencies on touch screens [15]. With an experimental device having only 1ms of latency, they showed that participants were able to perceive latencies only 6ms above the baseline on average, with some participants reaching 2ms. In another study, the ability to perceive latency was found to be strongly dependent on the task [14]. Using a stylus, the perceivable latency was found to be smaller when dragging a small square (2ms) compared to a big square (6ms) and to scribbling (40ms). This higher perceptual threshold for more demanding tasks was confirmed by Annett et al.: the threshold was found around 53ms for writing or drawing [1]. Recent work from Deber et al. [5] confirms that the latency is more perceivable with direct interaction compared to indirect, and they show that small improvements of latency (e.g. 8ms) are noticeable from a wide range of baseline latencies, particularly when dragging. These studies indicate that even though very small level of latencies are perceivable, it may or may not be considered an annoyance by users depending on their expectations from the system and the task being performed.

Rather than *explicitly* asking participants about their perception of latency, an *implicit* effect of latency can be observed by measuring users' performances in the execution of a common task such as target acquisition. This was studied by MacKenzie and Ware for regular mouse interaction [12]. By testing 8ms, 25ms, 75ms and 225ms of latency, they showed its strong negative effect on user's performance, with the movement time increasing by 63% and error rate jumping from 9% to 21% when latency increased from 8ms to 225ms. Users' performance also degraded between 25ms and 75ms of latency, but *not* between 8ms and 25ms. Ivkovic et al. ran a similar experiment with mouse pointing but for a central reticule-aiming task [6]. They observed performance degradation when varying latency from 164ms down to 41ms, but not between 41ms and 11ms. Although not studying exactly the same pointing tasks, these two studies considered together indicate that a threshold of the influence of latency on (indirect) mouse interaction is in the range [41ms-75ms]. Ivkovic et al. also demonstrated that latency could be efficiently counteracted by the use of a target-aware "sticky target" technique. With the widespread diffusion of direct-touch interaction through smart-phones and tablet computers, the influence of latency on users' performance came back as a major concern. Using the same 1ms latency hardware as Ng et al [15], Jota et al. studied users' performances for a dragging task [7]. Contrary to mouse interaction, the detrimental influence of latency was found at levels lower than 41ms. By doing a linear regression on their experimental data, the authors suggest that there may not be a floor effect and that users' performance may increase as long as latency decreases, at least down to 1ms. However, due to the constrained size of the experimental device, only "easy" dragging tasks were tested with the highest Fitts' Index of Difficulty (ID) at 2.58bit. Thanks to the availability of a larger touch device, our work extends these results to a more common range of IDs including "difficult" targets up to ID=6.3bit.

Predicting trajectories in HCI

Prediction has been used in several research efforts in an attempt to speed up target acquisition in pointing tasks. Asano et al. made a target prediction based on the peak velocity of the pointer [2]. Subsequent efforts studied more complex models of prediction: motion kinematics [8], neural networks and Kalman filters [4], and kinematic template matching [16]. In all these efforts, complex models are used to address the challenge of a long prediction: there are typically several hundreds of milliseconds between the date of the prediction, at the beginning of the pointing gesture, and the final target acquisition. The prediction is never accurate enough to automatically select the target, and final adjustments from the user are expected. The most recent efforts involve the learning of the model's parameters from example trajectories [4, 16]. Our work focuses on similar target acquisition tasks, although the approach is quite different: rather than guessing which target the user is pointing at, we improve the coupling between the user's finger and the virtual object, aiming at the ideal zero latency condition that would allow users to perform the task with optimal performances.

Xia et al. recently studied the use of a prediction as a means to reduce latency in direct touch systems, however they focused on tapping tasks, i.e. when the finger makes the first contact on the surface [20]. Using the hovering trajectory of the finger before it touches the surface, they predicted both the endpoint of the movement and the touch time by fitting a parabola on the current trajectory and using a linear extrapolation on the last phase of the gesture. The system was able to provide an immediate feedback when the user was tapping, virtually removing the latency of the system. Our work shares the same objective, but we focus on dragging instead of tapping.

Our approach finds its inspiration in the efforts to create visual immersion in Virtual Reality systems. Latency has long been acknowledged as a bottleneck for visual immersion: it is a source of simulator sickness [9] and may completely break the illusion of presence [19]. Continuous predictions of the user's head position and orientation have been used in various attempts to compensate latency. Lengthy predictions were used to deal with large latencies, typically in the range of 300ms [19]. However, as improvements in sensing and display have reduced systems' latency, more simple prediction models were used. LaViola showed that a prediction based on simple linear regressions performed as well as extended Kalman filters when predicting head pose for 50ms and 100ms [10]. More recently, LaValle et al. discussed the predictive methods used in the popular Oculus rift HMD [9]. The method predicts head orientation for 20ms and 40ms using only "a few milliseconds of data" from 1kHz sampling of angular velocity, and a simple constant acceleration model. The authors emphasize the requirements for a successful prediction: dense, accurate sensor data, and a limited length of the prediction. Our work follows the same path of using only the most recently sensed data, a simple prediction model, and a short prediction. We aim at discovering if and how the same convergence of fast hardware and accurate enough prediction can be successful. Compared to visual immersion, direct touch faces the challenge of touch sensors reporting positions rather than speed, and usually at a much lower sample rate. In addition, direct-touch interaction has different requirements and may be even more sensitive to latency, as hinted by the results from Jota et al. [7].

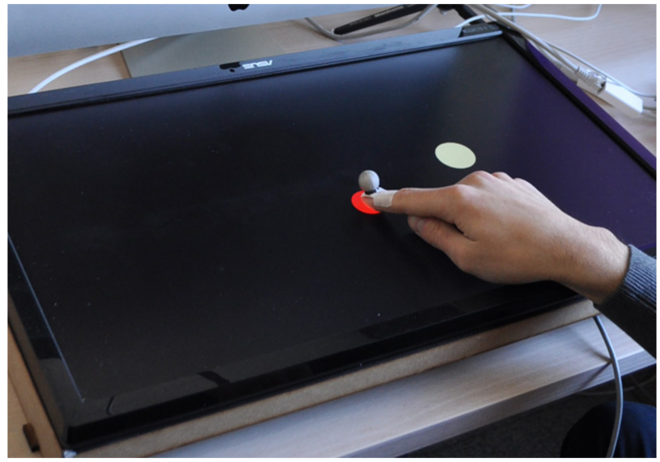

Figure 1. The experimental system in use during the experiment. The finger is localized by optical tracking using a marker. Users touch the red disc to bring it into the white target.

The soon to be released iOS 9 mobile operating system offers touch prediction services in order to reduce direct touch latency [18]. While the details of the predictive algorithm are not published, the system uses the 120Hz touch sampling of the iPad Air 2 to provide positions predicted 16.7ms in the future (i.e. one 60Hz display period). With the iOS rendering pipeline, the prediction theoretically reduces the latency to 1.5 display period (25ms). We follow the same continuous prediction approach to reduce latency, but we experiment with a larger interactive surface that is subjected to more visible lag due to faster dragging. We also experiment with longer predictions in an attempt to entirely compensate the system's latency. We finally provide an objective assessment of the prediction efficiency by measuring its effect on users' performance.

EXPERIMENTAL SYSTEM

A low latency direct-touch surface

Previous efforts indicate that users' performances in target acquisition are very sensitive to tracking errors: even a small gap created by 10ms of latency has a detrimental effect on performances [7]. We thus anticipated that the continuous prediction approach might not work for long predictions meant to compensate the high latencies of currently available commercial devices: for any prediction model, the longer the prediction, and the larger the predicted error. We thus built a specific touch device that allowed us to test various levels of latency starting at 25ms. However, contrary to the system used by Ng. et al [15], our system only relied on off-the-shelves hardware that did not deviate too far from the capabilities of standard direct touch hardware. In particular, our system used a standard monitor that offers a width of 52cm and a color display, and the software used a standard rendering pipeline on a common OS.

A 24inch Asus VG248E 1920x1080 LCD display without its foot and horizontally set on a desk was used as the direct-touch surface. The display was set in a "game mode" which disabled any processing on the video signal to provide fast response time. The display was driven at a refresh rate of 120Hz (period of 8.33ms). To get fast touch sensing on the display, we used two Natural Point Optitrack Flex-13 cameras for optical tracking. The tracking system reported the 3 degrees of freedom of reflexive markers at 120Hz. We calibrated the parameters of the display's plane in the cameras' 3D space by sliding a marker on the surface of the display. A spherical marker was attached to the participants' index finger, as illustrated on Figure 1. The 3D position of the marker is used to inform the system about the contact of the finger on the display and is then projected onto the display plane to get the position of the finger's projection on the display. By recording the sensed position of a stable marker put on the monitor, and repeating the operation in various locations on the monitor, we measured the precision of the sensing at less than 0.1mm (sub-pixel). The tracking software ran on a dedicated PC that sent marker positions to the main PC via the UDP protocol on a 1Gbit/s wired network. The round-trip time was measured below 0.5ms with low variability. The main PC was running Mac OS X 10.9.5 on a quad-core 3.4GHz processor and was equipped with an NVIDIA GeForce GTX 680MX graphic card. The system's software was custom written in C++ and used the OpenGL library for graphical rendering.

We assessed the end-to-end latency of our system by using the "high accuracy" approach introduced by Berard and Blanch [3]. Doing several measurements during the course of the experiments, we verified that the average latency was stable. In each measurement, latency averaged at 25ms and varied in the range [17ms-33ms]. From this variability, 8.33ms can be explained by the display refresh period. The remaining variability can be attributed to variable delays introduced by the system's components: computation of the 3D position of the marker on the tracking PC, communication between the tracking and main PCs, rendering of the scene and its communication to the display. We implemented artificial latencies by queuing finger motion events and releasing them only after a number of display cycles. We used 3 levels of latency in addition to the baseline: 42ms (+2 cycles), 58ms(+4 cycles) and 75ms(+6 cycles). The artificial latencies were controlled using the same procedure as for the baseline, and showed the same stability of the averages and the same variability around the averages.

Predicting finger trajectories

Jota et al. decomposed target acquisition trajectories in 3 phases: initial reaction, large ballistic motion, and "feedback-adjusted final adjustments" where the finger is inside the target area [7]. They found that the last phase was the most affected by latency. This could be explained by the fact that, with direct touch, the physical finger has no latency and allows users to control their gesture during the large initial motion. The lagging digital object is only required at the end of the gesture for its precise positioning within the target. To get a better understanding of this phenomenon, we studied twenty finger trajectories of different participants aiming at small distant targets (amplitude = 33.2cm, width = 0.42cm). We hand labeled the duration of the small final adjustments by looking at the speed profiles. We found that correction lasting less than 70ms were frequent and were often separated by even shorter motionless phases. We also observed that these corrections were quite erratic and thus appeared as unpredictable. This uncovers the challenge of a continuous prediction for target acquisition: compensating latency is most needed on the last phase of the acquisition gesture, which is irregular and rapidly evolving. It results that the amount of latency that can be compensated is bounded by the duration of these small adjustment motions: a system with 70ms of latency is only informed about the beginning of such short motion when the motion has already finished in the physical world; the system has no way to predict it.

These observations point to a highly reactive prediction model using only the most recent sensor data, similar to the most recent approach in visual immersion [9]. We considered using a constant acceleration parabolic model by fitting two parabolas (x and y) on the 3 most recent samples. To get a sense of the prediction's accuracy, we plotted a predicted trajectory alongside a no-latency "physical" trajectory that we simulated by removing a constant latency on each sample, i.e. translating a recorded trajectory on the time axis by the simulated latency. We used a simulated latency of 75ms to magnify the effect of the prediction. As illustrated on Figure 2(A), we observed that the instability of the parabolic fit interacted badly with the sensor noise and produced unstable predictions. To deal with sensor noise, we added more samples to the fit. However, any added sample is 8.33ms older than the previous one. The more sample added, and the less reactive the model becomes as it represents outdated evolutions of the trajectory. The phenomenon is revealed on Figure 2(B) with a 10 samples least-square optimized fit of the parabolic model: the prediction is much smoother, but at the cost of inertia that does not prevent strong under- and overshoots. We also experimented with a simpler constant speed linear prediction. The benefit of using a first order polynomial is that it only requires the two most recent samples for its estimation; hence it is very reactive. In addition, it only requires only one level of time integration for the position prediction, which results in lesser amplification of sensor noise. The result is illustrated on Figure 2(C, D) for a 75ms and a 25ms prediction.

We thus focused the scope of our study on the simple linear prediction model and with prediction lengths at or lower than 75ms. We attempted to entirely counteract the system's latency: we made a prediction of Xms for a system that has an average latency of Xms. In our model, we approximated the latency of the system as a constant l, measured with the "high accuracy" method of Berard and Blanch [3]. Although we observed some variability (±8ms) around the average, we had no way to estimate the exact latency at each sample. This situation is representative of most current direct touch system running on standard time-sharing OSes. We observed that the Optitrack sampling rate was very stable and thus we approximated the sample period as a constant $\Delta t = 1000/120 = 8.33ms$.

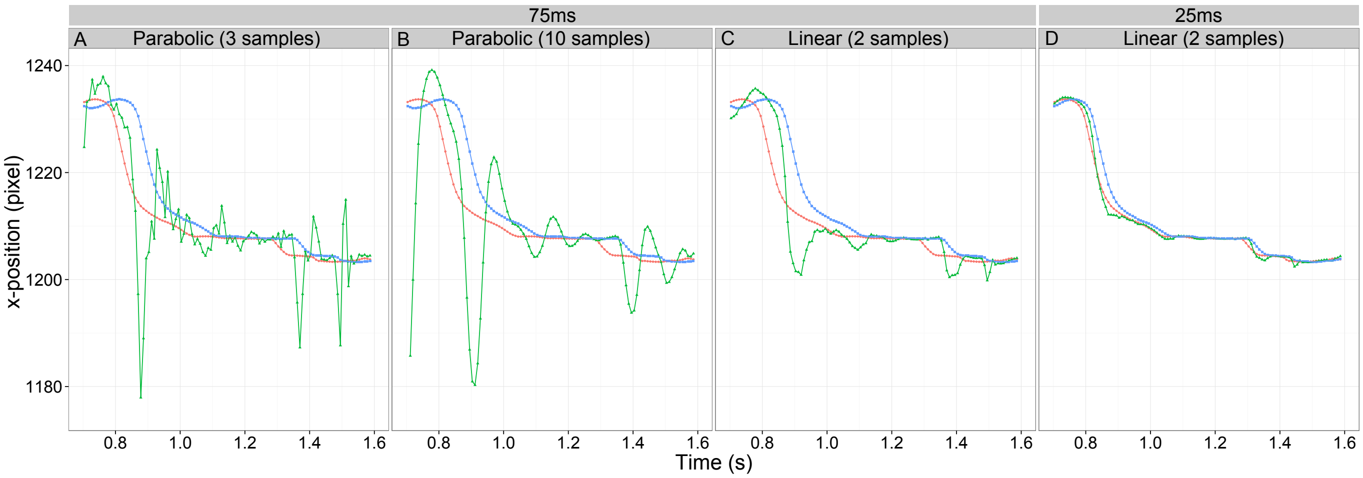

Figure 2. Final phase of a target acquisition. Simulated physical trajectory (red line), raw trajectory with lag (blue line) and predicted trajectory (green line). Prediction at 75ms of latency with a constant acceleration parabolic model fit on 3 samples (A) and on 10 samples (B), and with a constant speed linear model fit on 2 samples (C). Prediction with the linear model at 25ms of latency (D). Only the x-position is shown for a mostly horizontal trajectory.

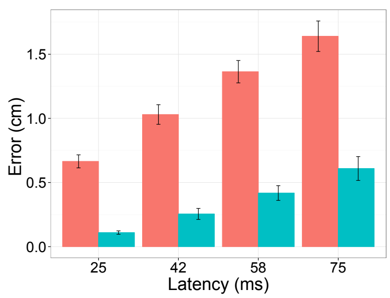

Figure 3. Average error per sample between simulated no-latency trajectories and lagging trajectories (in red) or predicted trajectory (in blue).

Finally, the *predicted* 2D position of the finger $\mathbf{\hat{x}_i}$ at display cycle number i was computed as:

$$\mathbf{\hat{x}_i} = \mathbf{x_i} + l * \mathbf{\hat{s}_i} \qquad \mathbf{\hat{s}_i} = \frac{\mathbf{x_i} - \mathbf{x_{i-1}}}{\Delta t} \tag{1}$$

where the $\mathbf{x}$ are the *observed* finger positions, $\mathbf{\hat{s}}$ is the speed estimation, l is the measured system's latency and $\Delta t = 8.33ms$. One of the benefits of this simple model is its user- and target- agnostic nature; hence it can be implemented with minimal effort in most direct-touch systems.

To get a sense of the accuracy of the prediction, we computed the average position error per sample for raw and for predicted trajectories. The error was computed by comparison to no-latency "physical" trajectories simulated as described above. These errors were averaged on 7428 trajectories recorded during our experiments. The results are illustrated on Figure 3. At 25ms of system latency, the average error is lowered from 0.66cm down to 0.11cm, an 84% improvement. At 75ms, it is lowered from 1.6cm down to 0.61cm, a 63% improvement. Although this gives a very positive picture of the global prediction's accuracy, looking at trajectories in details gives another picture (i.e. Figure 2(C)): the prediction introduces undershoots in accelerating phases of a motion, and overshoot in decelerations. This is particularly visible at 75ms of system latency, and in the final adjustments of a target acquisition. In this particular case, the challenge of predicting rapidly evolving corrections from late data appears distinctly, with the prediction introducing more perturbation than the lagging trajectory.

A user study was thus required to get an objective picture of the benefits of the predictive approach on users' performance.

USER EXPERIMENTS

The main objective of this user experiment was to evaluate if the predictive approach could effectively counteract the negative effect of latency on target acquisition tasks in term of users' *performance*, but also in terms of users' *preferences*. Doing so, we wanted to find which levels of latencies were tractable with the predictive approach. A secondary objective was to reproduce Jota et al.'s results [7] assessing the *detrimental effect of latency* on users' performance in direct-touch target acquisitions, and extending it to higher Fitts' ID.

The participants were 24 volunteers from our academic environment (average age=27.4, range 22 to 39, 8 women). After a short training session (10-15min), the participants performed the first experiment involving target acquisitions with varying Fitts' IDs for about 25min. They were then invited to perform the second experiment in order to express their preferences when dragging a virtual object with or without prediction.

EXPERIMENT 1: EFFECT OF LATENCY AND PREDICTION ON TARGET ACQUISITION PERFORMANCES

Task

Participants' task was to land their finger on a red disc *object* (1.38cm radius) and drag it on top of a white disc *target*. The acquisition of the target was successful only when the object was totally inside the target. Therefore, in the remainder of the paper, we refer to the width of the targets in terms of Fitts Index of Difficulty (ID): the target width is the difference in diameter between the object and target discs. The graphical display is illustrated on Figure 1. Participants validated the acquisition by releasing their control on the object, i.e. by lifting their finger from the surface. On finger release, the target turned green or red depending on the success or failure of the acquisition. After a random delay in the range [0.2s-1.2s] to prevent participants' anticipation, the target of the next trial appeared. Between two successive trials, the object remained at the location where the user released it. We asked participants to be as fast as possible while trying to limit their error rate at "around one mistake between each pause", which corresponds to 5%. All targets were located on a horizontal line vertically aligned with the middle of the screen to avoid the occlusion of targets by the hand. Every 20 trials, the message "you can take a break" was displayed. When they were ready, participant resumed the experiment by grabbing the object again. We recorded the trajectory of the object.

Design, measurement and analyses

Our design involved three main factors:

- PREDICTION. Two levels: $false$ (no prediction) or *true* (with prediction). When *true*, the prediction attempted to entirely counteract the latency of the system: we did not study partial compensations in order to limit the number of experimental conditions;

- LATENCY. Four levels: 25ms, 42ms, 58ms and 75ms. The smallest latency was defined by the best performance of our system. For the upper bound, 75ms was a typical value of the duration of the shortest corrective motions at the end of target acquisitions.

- ID. Four levels: Fitts IDs are provided in bits and distances in cm.: 2.32 (d=11, w=2.77), 3.64 (d=22.14, w=1.94), 5.04 (d=22.14, w=0.69), 6.34 (d=33.2, w=0.42). These cover big and small targets at large and short distances.

Overall, the experiment included 32 conditions (2 PREDICTION x 4 LATENCY x 4 ID) and 20 repetitions for each condition, which amounted to 15360 trials (24 participants x 32 conditions x 20 trials). The order of presentation of PREDICTION and LATENCY was balanced across participants as a means to equilibrate a potential order effect. Within each block of PREDICTION x LATENCY, the presentation of the 80 targets (4 IDs x 20 repetitions) was randomized, but the same order was used in all blocks and all participants. Presentation of conditions was ordered by PREDICTION first, then LATENCY.

In our approach, touch detection was achieved by thresholding the altitude of the finger above the surface, with a hysteresis to avoid instabilities. However, the finger 3D pose varies depending on the touch location: the further away the user is touching, the more horizontal the finger. This introduced variability in the altitude of the marker, and required that the threshold be defined for the pose that yielded the highest altitude. The consequence was that our touch detection was not as sensitive as on a typical capacitive or FTIR touch surfaces. As a result, the virtual object often remained under control of the finger for a very short moment after the finger was lifted from the surface, and this frequently generated unintended object's motions. Parasitic motion on release exists with any kind of touch technology; however, our experimental optical tracking approach clearly inflated the problem and increased the acquisitions' failure rate. As this may have biased our participants to slow down, we verified that this was not the case by looking at the evolution of the velocity profiles during the experiment. They showed no sign of a slow down, but rather a tendency of increasing speeds that we attributed to the participants' training to the task.

We then removed, for each subject and condition, trials with duration at more than 1.5 interquartile above the upper quartile or below the lower quartile. These outliers amounted to 3.125% of all trials.

Based on these pre-processing, we then computed, for each subjects and experimental condition:

- *mean_time* as the mean target-acquisition time defined as the time span between the touch and release of the finger;

- *err_rate* as the ratio of failed trials over the number of analyzed trials. The acquisition was considered as failed when the object was not entirely within the target on release;

- the throughput TP as defined in Soukoreff and MacKenzie [17], expressed in bits per second.

The effects of PREDICTION, LATENCY and ID and their interaction were evaluated with within-subjects ANOVAs. Paired Student's t-tests were computed to test for the effects of planned comparisons related to our hypotheses: (1) a negative effect of latency on users' performance and (2) a positive effect of prediction for lower latencies.

Results

Effect of latency on users' performances

In order to compare our study with previous work [12] and to test the effect of latency on a larger range of ID than in Jota at al. [7], we analyzed the effect of LATENCY and ID on *mean_time*, *err_rate* and TP. This analysis concerns only the subset of data in the PREDICTION=$false$ condition. As previous work didn't include a condition with prediction in their design, we first added the presentation order of PREDICTION levels (*true* first vs. *true* second) as a between subject factor to the ANOVAs. The effect of order on *mean_time*, *err_rate* and TP was not significant and didn't significantly interact with the effect of LATENCY and ID. We thus merged the two groups for further analyses. We refer to this subset of the experimental data as the "no prediction" dataset. The effects of ID was always highly significant, the following analyses focus on the effect of latency and its interaction with IDs.

mean_time significantly increases with LATENCY ($F_{3,69} = 13.6$, $p < .0001$, Figure 4) between 25ms and 42ms (+61ms, $t(23) = 2.9$, $p = .017$) and between 58ms and 75ms (+80ms, $t(23) = 3.1$, $p = 0.005$), the change between 42ms and 58ms was not significant (+4ms, $t(23) = 1.3$, $p > .2$). The effect of LATENCY on *mean_time* was also larger for higher IDs

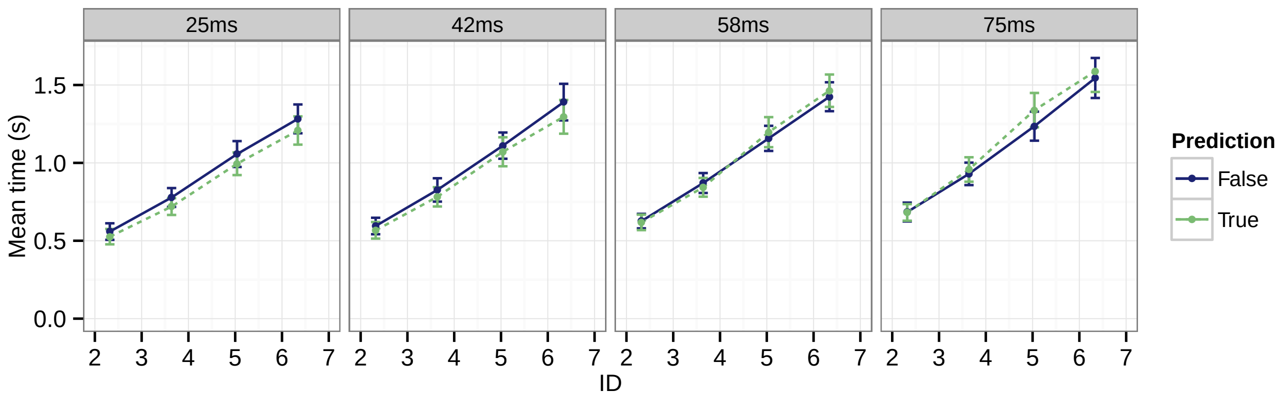

Figure 5. Mean target-acquisition time depending on target ID, latency, and prediction. Error bars give the 95% confidence intervals of the mean across participants.

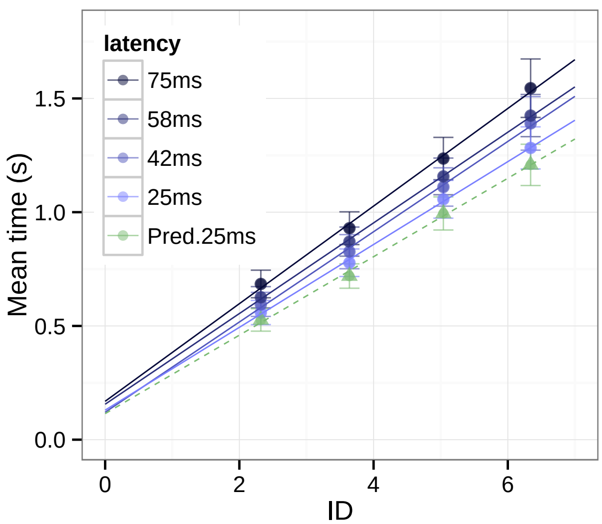

Figure 4. Effect of system latencies on the mean target-acquisition time when no prediction is used. The condition with prediction at 25ms of latency is added on the figure for comparison. The points are the mean values across 24 participants and error bars give the 95% confidence intervals of the means. The lines are linear fits of the data in each condition.

than smaller ones ($F_{9,207} = 4.7, p < .0001$): the slopes of a linear regression of *mean_time* according to ID increased with LATENCY ($F_{3,69} = 7.3, p < .001$) between 25ms and 42ms (+0.016, $t(23) = 2.4, p = .023$) and between 58ms and 75ms (+0.015, $t(23) = 2.6, p = 0.015$), the slope change between 42ms and 58ms was not significant ($t(23) = 0.1, p > .9$). TP decreased with increasing LATENCY ($F_{3,69} = 21.0, p < .0001$), changes were significant between 25ms and 42ms (-0.22bit/sec, $t(23) = 2.1, p < .05$), between 42ms and 58ms (-0.34bit/sec, $t(23) = 3.0, p < .006$) and between 58ms and 75ms (-0.28bit/sec, $t(23) = 3.2, p < .005$). There was no global effect of LATENCY on *err_rate* ($F_{3,69} = 0.99, p > .4$), and no interaction between LATENCY and ID ($F_{9,207} = 1.7, p > .1$).

Effect of the prediction on users' performance

The effect of the prediction on users' performances was analyzed using the full dataset. The individual effects of ID and LATENCY were comparable to the ones observed in the "no prediction" dataset. We thus only focus on the effects related to PREDICTION.

The global effect of PREDICTION on *mean_time* was not significant ($F_{1,23} = 0.4, p > .5$), due to different effects of PREDICTION according to LATENCY and ID ($F_{9,207} = 4.0, p < 0.0001$, Figure 5). PREDICTION appears to have a negative effect at 75ms, a neutral effect at 58ms and a positive effect at 25ms and 42ms. Independent within subjects ANOVAs with PREDICTION and ID as factors were ran for each of the LATENCY conditions to better understand these interactions. At 25ms of latency, the average *mean_time* across IDs and participants was significantly reduced by 6.2% with the use of the prediction (-57ms, $F_{1,23} = 6.4, p = .02$). A non-significant tendency was observed at 42ms ($F_{1,23} = 3.9, p = .06$) but no effect of PREDICTION was found at 58ms and 75ms ($F_{1,23} < 1.1, p > .3$ in both cases). The effect of PREDICTION didn't clearly depend on ID at 25ms ($F_{3,69} = 1.8, p = .16$), while it tended to increase with higher IDs for the three higher levels of LATENCY ($F_{3,69} > 3.2, p < 0.03$).

err_rate was significantly affected by PREDICTION ($F_{1,23} = 41.6, p < .0001$, Figure 6). However, PREDICTION had different effects on *err_rate* according to ID ($F_{3,69} = 29.9, p < .0001$) and LATENCY ($F_{3,69} = 12.7, p < .0001$). Independent ANOVAs for each levels of LATENCY showed that, at 25ms, *err_rate* was globally not significantly affected by PREDICTION ($F_{1,23} > 3.3, p > .0.08$). PREDICTIONS effects on *err_rate* were significant for 42ms, 58ms and 75ms ($F_{1,23} > 18.5, p < .001$).

The effect of LATENCY on TP was significantly different according to PREDICTION ($F(3,69) = 9.3, p < .001$). The prediction increased throughput at 25ms (+0.15) but not significantly ($t(23) = 1.4, p > .17$), at 42ms, the throughput remains stable. The prediction decreased throughput at 58ms and 75ms ($t(23) > 2.5, p < .03$).

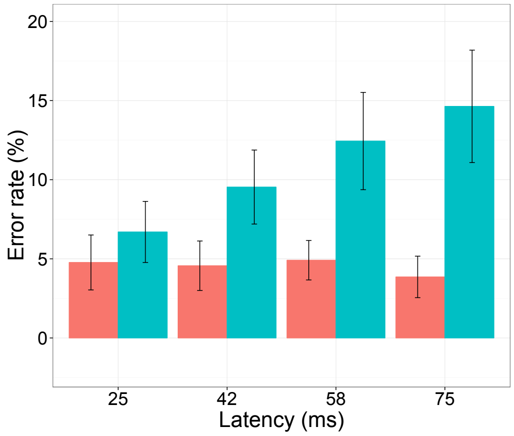

Figure 6. Mean error rates across participants averaged across IDs, and 95% confidence intervals of the mean. Left side red bars represent the "no prediction" condition, right side blue bars the "with prediction" condition.

EXPERIMENT 2: EFFECT OF LATENCY AND PREDICTION ON USER'S PREFERENCES

Task and design

Performance is not the only criteria to design a good user-interaction: users' comfort is another important aspect. We evaluated users' preference for the interaction with or without prediction. We asked participants to interact with a draggable virtual disc in two conditions A and B. Both conditions had the same amount of latency, one was using the prediction and the other was not. As in experiment 1, the prediction was set to entirely compensate the systems' latency. Participants were not told which of A or B was using the prediction and the assignment of prediction to A or B was randomized across trials. Participants could switch between A and B by pressing the space key on a keyboard as many times as they wanted. They ended each trial at their convenience, by expressing which of A or B they found "the most comfortable". They were asked to always provide an answer, even when the difference between the 2 conditions was difficult to perceive (mostly at low latencies).

As the preference may depend on the type of task performed, we split the sessions in two parts. In the four first trials, the 4 latencies were tested with no specific task: participants could explore freely the dragging interaction with the disc. In the next 16 trials, participants were asked to do target acquisition tasks for the 4 IDs and 4 latencies. The order of presentation of the latencies was balanced across participants. For a given latency, IDs were presented in the following order: 2.32, 5.04, 3.64, and 6.34.

Results

Figure 7 illustrates the results of the preference experiment. We tested each preference score with a chi-square to check if the null hypothesis of a random choice could be rejected. When asked to perform target acquisitions, participants preferred the "no prediction" condition in the three highest latencies ($\chi_2(1,96) > 13, p < 0.001$), but the random choice hypothesis could not be rejected at 25ms ($\chi_2(1,96) = 0.67, p > 0.41$). When freely interacting, 17 out of 24 participants (70.8%) preferred the condition with prediction at 25ms latency ($\chi_2(1,24) = 4.17, p < 0.05$), while a random choice could not be rejected for the three highest latencies ($\chi_2(1,24) < 1.50, p > 0.22$).

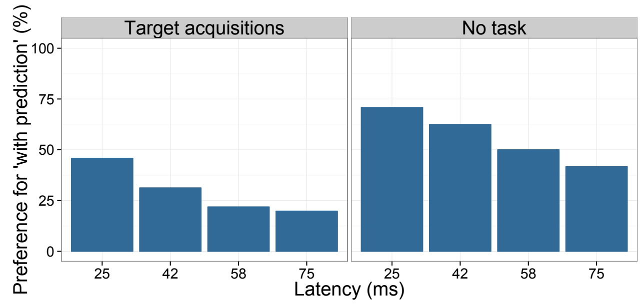

Figure 7. Proportion of participant preferring the "with prediction" condition during the subjective experiment.

DISCUSSION

Effect of latency on users' performances

The analysis of the "no prediction" dataset replicates and confirms previous work, which showed the strong detrimental effect of latency on users' performance in target acquisition tasks [7, 12]. Latency acts as a multiplicative factor of the index of difficulty: as latency increases, the throughput decreases and the slopes of the performance curves in Figure 4 increase. Performance continued to improve as latency was reduced to our lowest achievable value of 25ms. This is coherent with the results from Jota et al. [7], which we extended to the acquisition of targets IDs greater than 2.6bit up to 6.3bit. The first outcome of this replication of previous work was to validate our system and our experimental design. It also provided an indication of the expected users' performance depending on latency, which we discuss later in the paper.

Counteracting latency with a continuous prediction

Using prediction with latencies *above 42ms* did not yield better users' performances overall. Performances with prediction were the lowest for the two targets with the highest IDs. The small sizes of these targets (0.7cm and 0.4cm) required precise control and thus several adjustments in the final phase of the acquisition. Our results corroborate previous studies that indicate that latency has its strongest negative influence on the final adjustment phase of target acquisitions [7]. In addition, the study reveals that the continuous prediction approach has a rather low upper bound on the amount of latency that it can compensate, at least when used to perform tasks that require precise positioning. 42ms may appear as a small prediction length compared to typical prediction lengths used in endpoint prediction, which often exceed 100ms. However, endpoint prediction is essentially based on the initial motion of a target acquisition, which is much more regular (and thus predictable) than the final adjustment motions. In addition, endpoint prediction is generally used for target identification, which has a lower accuracy requirement compared to latency compensation.

By including target acquisitions of high IDs in our experiments, we probably have selected tasks that are amongst the most sensitive to prediction errors. Other direct-touch tasks, such as scrolling in lists or navigating in maps typically don't require sub-centimeter precision. As a consequence, our results should be considered as a worst-case scenario of the use of a continuous prediction in direct-touch interaction. In addition, the subpar touch-release detection of our experimental device should also be taken into consideration when considering the measured error rates: larger than average final parasitic motions probably increased the number of errors. Better touch release detection on common capacitive or FTIR surfaces would provide lower error rates in all the conditions represented on Figure 6. However, as the prediction amplifies the parasitic motions on release, the error rates with prediction should see greater improvements, hence the gap in error rates with and without prediction is expected to decrease.

Even with high precision target acquisitions, the prediction did improve users' performance when the latency of the system was at 25ms. To get a sense of the scale of the improvement we used the "no-prediction" dataset: in Figure 4, the improvement from 41ms to 25ms is similar to the improvement provided by the use of prediction at 25ms. As in Jota et al. [7], we computed a linear regression of the average task accomplishment time across IDs and participants, depending on latency. We found a good fit ($R^2 = 0.984$), and used it as an indicator of the performance that could be expected depending on the latency of the system. We identified the latency of a hypothetical device that would yield an average task accomplishment time of 0.795s, which corresponds to the performances of participants using prediction at 25ms. The linear regression indicates that the prediction allowed users to perform as well as with a 9ms of latency system, a 64% improvement over 25ms. In summary, assuming that the baseline latency of the system is low enough, the experiment demonstrated that the continuous prediction approach can bring the apparent system latency much closer to the ideal no latency, and this allows users to improve their performances.

Users' preferences

The results of the users' preferences experiment bring another light on a similar picture: participants favored the prediction only for the lowest levels of latencies, and only when we did not ask them to perform a task that required notable accuracy. When free to interact as they wanted, participants tended to make slow and smooth movements. This favored the prediction since there was no sudden speed variation and the amplitude of the prediction error remained small. In this context, the preference of participants was largely in favor of the prediction at 25ms of latency as illustrated in Figure 7 (right).

When asked to perform target acquisitions, participant preferred the "no-prediction" condition overall, as illustrated in Figure 7 (left). To justify their choice, participants commented that they felt "instability" and "absence of control" of the dragged object. In addition, the parasitic motion generated by our system may have played as a negative bias against the preference for the prediction. At 25ms of latency, some participants expressed difficulties in perceiving a difference between the two conditions. This is coherent with the non-rejection of the null hypothesis in this condition: a random choice is expected from undistinguishable conditions. Considered in regard to the performance increase shown in the first experiment, this result indicates that even though users may not perceive the difference, reducing the virtual object's lag allows them to be more efficient in precision tasks.

Applicability to current hardware

Current commercial and experimental devices usually have baseline latencies well above 25ms. However, reaching 25ms of latency does not seem out of reach or to require significant hardware improvements. 120Hz displays are widely available thanks to the push for stereo 3D. The iPad Air 2 already senses touch at 120Hz, and recent research work demonstrated touch sensors running at 4000Hz and with 0.04ms of latency [11]. USB3 cameras offer high spatial and temporal sensing for the implementation of large FTIR interactive surfaces (e.g. 2048x1088pixels at 170fps). A 120Hz display has 0ms of latency just after refresh, 8.33ms just before the next refresh, hence it introduces 4.17ms of average latency to the system. This leaves more than 20ms of additional average latency that may come for the other sources such as the sensing, the graphical rendering, and the synchronization overheads. The time needed to render the graphical feedback can vary greatly depending on the complexity of the scene, but for simple 2D user interfaces, this is typically done in a few milliseconds by current graphic cards[1]. With a 60Hz display refresh rate, 17ms of average latency are still available once the 8.33ms of average display latency are removed.

Our study should thus provide a strong incentive for touch system and application designers to maintain the non-display sources of latency in the range of 20ms. When this is the case, the continuous prediction offers a low effort instrument to provide users with a close to ideal system. Contrariwise, reaching 9ms of system latency using a pure hardware approach appears to be more difficult and energy inefficient.

However, the relevance of the predictive approach could be more general than the single case of 25ms latency systems. This points to further studies, as detailed below.

Future work

Deber et al. recently demonstrated that small improvements of latency, such as 16ms, were easily perceived by users in a direct touch dragging task on systems with various baseline latencies ([8.3ms-167ms]) [5]. Combined with our study, this creates strong expectations for a follow up study on the use of a continuous prediction for a *partial* compensation of latency on high latency systems.

In addition, ways to improve the continuous prediction and allow longer prediction lengths should be investigated. We observed that adding touch samples to average sensor noise had the negative effect of adding inertia to the prediction. Increasing the *sample rate*, however, allows increasing the number of samples without looking further in the past. Using high touch sampling rate, such as 4kHz demonstrated by

[1]The very simple scene of our system was rendered in less than 2ms, which we attributed mostly to OpenGL synchronisation overheads.

Leigh et al. [11], should be a promising approach to allow the use of higher degrees of freedom prediction models that remain reactive. A more ambitious path of improvement will be to study if some regularity can be found in the final adjustment motions of target acquisition. If it is the case, this could used to inform a specific prediction model.

CONCLUSION

In this paper, we have provided the first objective evaluation of the benefits of a continuous prediction to reduce latency and improve users' performances in direct touch systems. We showed that the length of the prediction is strongly constrained by the fast adjustments that make the end of target acquisition gestures. However, the experiment revealed that the continuous prediction approach could compensate a large part of latency's negative effect on target acquisition performances when running on a 25ms latency system. These results contribute to the quest of zero-latency direct touch.

ACKNOWLEDGEMENT

This work has been partially supported by the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01).

REFERENCES

1. Michelle Annett, Albert Ng, Paul Dietz, Walter F. Bischof, and Anoop Gupta. 2014. How Low Should We Go?: Understanding the Perception of Latency While Inking. In *Proceedings of Graphics Interface (GI)*.

2. Takeshi Asano, Ehud Sharlin, Yoshifumi Kitamura, Kazuki Takashima, and Fumio Kishino. 2005. Predictive interaction using the delphian desktop. In *Proceedings of ACM UIST*. 133–141.

3. François Bérard and Renaud Blanch. 2013. Two Touch System Latency Estimators: High Accuracy and Low Overhead. In *ACM ITS*. ACM, 241–250.

4. Pradipta Biswas, Gokcen Aslan Aydemir, Pat Langdon, and Simon Godsill. 2013. Intent recognition using neural networks and Kalman filters. In *Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data*. Springer, 112–123.

5. Jonathan Deber, Ricardo Jota, Clifton Forlines, and Daniel Wigdor. 2015. How Much Faster is Fast Enough?: User Perception of Latency & Latency Improvements in Direct and Indirect Touch. In *Proceedings of ACM CHI*.

6. Zenja Ivkovic, Ian Stavness, Carl Gutwin, and Steven Sutcliffe. 2015. Quantifying and Mitigating the Negative Effects of Local Latencies on Aiming in 3D Shooter Games. In *Proceedings of ACM CHI*.

7. Ricardo Jota, Albert Ng Ng, Paul Dietz, and Daniel Wigdor. 2013. How Fast is Fast Enough? A Study of the Effects of Latency in Direct-Touch Pointing Tasks. In *ACM CHI*. ACM.

8. Edward Lank, Yi-Chun Nikko Cheng, and Jaime Ruiz. 2007. Endpoint prediction using motion kinematics. In *Proceedings of ACM CHI*. ACM, 637–646.

9. Steven M LaValle, Anna Yershova, Max Katsev, and Michael Antonov. 2014. Head tracking for the Oculus Rift. In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 187–194.

10. Joseph J. LaViola. 2003. Double Exponential Smoothing: An Alternative to Kalman Filter-based Predictive Tracking. In *Proceedings of the Workshop on Virtual Environments 2003 (EGVE '03)*. ACM, New York, NY, USA, 199–206.

11. Darren Leigh, Clifton Forlines, Ricardo Jota, Steven Sanders, and Daniel Wigdor. 2014. High rate, low-latency multi-touch sensing with simultaneous orthogonal multiplexing. In *Proceedings of ACM UIST*. ACM, 355–364.

12. I Scott MacKenzie and Colin Ware. 1993. Lag as a determinant of human performance in interactive systems. In *Proceedings of ACM CHI*. 488–493.

13. Michael Meehan, Sharif Razzaque, Mary C Whitton, and Frederick P Brooks Jr. 2003. Effect of latency on presence in stressful virtual environments. In *Virtual Reality, 2003. Proceedings. IEEE*. IEEE, 141–148.

14. Albert Ng, Michelle Annett, Paul Dietz, Anoop Gupta, and Walter F. Bischof. 2014. In the Blink of an Eye: Investigating Latency Perception During Stylus Interaction. In *ACM CHI*. ACM, 1103–1112.

15. Albert Ng, Julian Lepinski, Daniel Wigdor, Steven Sanders, and Paul Dietz. 2012. Designing for low-latency direct-touch input. In *Proceedings of ACM UIST*. ACM, 453–464.

16. Phillip T Pasqual and Jacob O Wobbrock. 2014. Mouse pointing endpoint prediction using kinematic template matching. In *Proceedings of ACM CHI*. ACM, 743–752.

17. R. William Soukoreff and I. Scott MacKenzie. 2004. Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI. *International Journal of Human-Computer Studies* 61, 6 (2004), 751 – 789.

18. Peter Tsoi and Jacob Xiao. 2015. *Advanced Touch Input on iOS*. Technical Report. Apple Inc.

19. Jiann-Rong Wu and Ming Ouhyoung. 2000. On latency compensation and its effects on head-motion trajectories in virtual environments. *The visual computer* 16, 2 (2000), 79–90.

20. Haijun Xia, Ricardo Jota, Benjamin McCanny, Zhe Yu, Clifton Forlines, Karan Singh, and Daniel Wigdor. 2014. Zero-latency tapping: using hover information to predict touch locations and eliminate touchdown latency. In *Proceedings of ACM UIST*. ACM, 205–214.