How Much Faster is Fast Enough? User Perception of Latency & Latency Improvements in Direct and Indirect Touch

Jonathan Deber [1,2] Ricardo Jota [1,2] Clifton Forlines [1,2] Daniel Wigdor [1]

[1] Dept. of Computer Science, University of Toronto
Toronto, ON, Canada
{jdeber, jotacosta, daniel}@dgp.toronto.edu

[2] Tactual Labs
New York, NY, USA
cliff.forlines@tactuallabs.com

ABSTRACT

This paper reports on two experiments designed to further our understanding of users' perception of latency in touch-based systems. The first experiment extends previous efforts to measure latency perception by reporting on a unified study in which direct and indirect form-factors are compared for both tapping and dragging tasks. Our results show significant effects from both form-factor and task, and inform system designers as to what input latencies they should aim to achieve in a variety of system types. A follow-up experiment investigates peoples' ability to perceive small improvements to latency in direct and indirect form-factors for tapping and dragging tasks. Our results provide guidance to system designers of the relative value of making improvements in latency that reduce but do not fully eliminate lag from their systems.

INTRODUCTION

Interface latency – the time interval between a user's action and the system's response to that action – is inherent in any computer system; input sensors must be sampled, computations performed, graphics generated, and displays updated. Existing commercial touchscreen devices have latencies that range between 50 and 200 ms [19]. In an ideal world, system designers would keep this latency below the threshold that is detectable by the human visual system, making the experience indistinguishable from a truly latency-free system. However, despite vast increases in the performance of computer systems, latency remains an ever-present blemish on the user experience.

Latency has been the subject of a large body of research, as it has a fundamental impact on the feel of a system and on user performance in pointing tasks. Early work provided guidelines for both direct and indirect pointing tasks [16] in the 100–1000 ms range. Performance-based measurements on indirect input devices showed little improvement below 75 ms [14]. More recent work has focused on direct touch user interfaces, in which the touch surface and display are co-located [8,18,19]. As a whole, this body of work paints a conflicting picture, with results indicating that the minimal perceivable latency rests anywhere between 2 ms and 100 ms. We argue that this discrepancy is due in large part to the various form-factors used in the previous work (indirect vs. direct) as well as the different input tasks (tapping vs. dragging). One of the goals of the present work is to help clarify the picture by conducting an experiment in which the relationship between direct and indirect input for tapping and dragging tasks is explored. The results of this work shed light on the likely sources of discrepancy in the literature and inform system designers about the minimal perceived latency for different touch-enabled systems.

Another shortcoming of the previous research in this space is that it focuses exclusively on identifying the minimum perceivable latency for different input tasks. A system designer reading this research would know the ideal latency to reach in order to deliver a perceptually "latency-free" system; however, they are given no guidance as to the value of making improvements to the latency of their device that do not reach this ultimate goal. In this paper, we also present a second experiment in which we aim to investigate people's ability to perceive improvements to latencies that fall well above this zero lower bound. Our results outline the value of improving latency by varying amounts in modern-day direct and indirect touch-based systems.

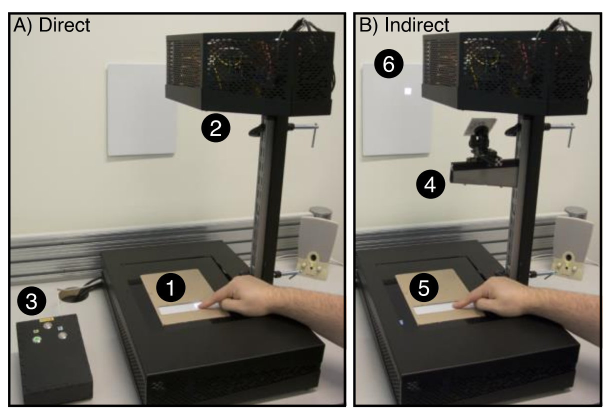

Figure 1. Experimental setup. A) In Direct mode, participants touch the surface (1) and output is produced by the downward-facing projector (2). After seeing a pair of latencies, participants indicate the faster of the two using the control box (3). B) In Indirect mode, the mirror assembly (4) redirects the projector's beam. The participant still touches the surface (5), but the output is displayed on the wall (6).

CHI 2015, April 18 – 23, 2015, Seoul, Republic of Korea

ACM 978-1-4503-3145-6/15/04…$15.00
http://dx.doi.org/10.1145/2702123.2702300

RELATED WORK

Latency, defined as the "delay between input action and the output response" [14], is an unavoidable part of any system, resulting from a variety of factors: reading the sensor(s) in an input device and transferring that data to the system, processing the input and performing application logic, rendering the updated UI, and waiting for the display to refresh its current image. The effects of latency on user interaction have been studied in previous works. Much of the prior work addresses *system latency*; they operate on the assumption of an existing baseline system latency and introduce additional latency as a factor. Techniques, such as those presented by Steed [25] and Kaaresoja et al. [9], can be used to measure baseline system latency. Researchers have also examined the point of subjective simultaneity (the separation at which two distinct stimuli are perceived as simultaneous) for haptic and visual stimuli [27], although most studies have not focused on touch input due to hardware limitations.

It is understood that latency affects how humans perceive virtual environments. Allison et al. studied the effect of latency in augmented reality, noting that latency degrades the illusion of stability – a major fault for an interactive system [1]. Nelson et al. reported that latencies of 50 and 100 ms impacted the ability to visually follow a virtual object with a head-mounted display [17]. Meehan et al. found that increased latencies of 50 and 90 ms reduced users' sense of presence in virtual environments [15].

We are particularly interested in the latency of input devices, and next examine work on indirect and direct systems.

Latency in Indirect Input Systems

Indirect systems separate input and output regions and require a pointing tool such as a mouse, touchpad, or stylus to provide information to the computer.

So et al. introduced latency (0, 55, 110, 220, and 440 ms) to indirect pointing, and found the effect was correlated with the width of a target and latency [24]. Pavlovych et al. tested mouse input for a targeting task with latencies above 20 ms and found that error rates increase significantly when latency rises above 110 ms [20,21,22]. Teather et al. observed that adding latency to a mouse and to a 3D tracker significantly impacted device performance; adding 40 ms to the system baseline latency affected performance by 15% [26]. Ware et al. studied the effect of latency for reaching tasks in 3D scenarios, and found that latency between 70 and 800 ms affects performance [28]. Ellis et al. found that latencies between 100 and 500 ms significantly degraded performance of path tracing tasks, and that users could distinguish latencies as low as 33 ms [5,6]. None of these studies examined touchpad-based interaction.

The disparity of the aforementioned results are likely due to the different devices being tested, supporting the suggestion that latency effects are task- and device-dependent.

Latency in Direct Input Systems

Unlike indirect input systems that separate the input device and the display surface, direct input devices (e.g., touchscreens) have no intermediary; the user inputs directly on the display surface. As with indirect systems, direct touch input devices also suffer from the effects of latency.

Anderson et al. conducted a qualitative study with users performing touchscreen tasks (e.g., web browsing and eBook reading) to determine the level of latency users find to be "acceptable" [2]. A delay above 580 ms was deemed unacceptable to the users, but it was noted that the experimental tasks were relatively brief (zooming, panning, and page-turning) suggesting that latency might be tolerable for longer tasks. Ng et al. studied the user perception of latency for touch input. For dragging actions, users were able to detect latency levels as low as 6 ms [19]. Jota et al. also studied touch input, and found that dragging task performance is affected if latency levels are above 25 ms, and that users are unable to perceive latency in response to tapping that is less than 24 ms [8]. Kaaresoja et al. looked at the visual perception of latency in physical buttons, and found the lower threshold of perception was 85 ms, but that the perceived quality of the button declined significantly for latencies above 100 ms [10].

Direct input systems that rely on a stylus, instead of touch, have also been studied. Ng et al. [18] focused on latency perception limits, and reported perception limits of 2 ms for a dragging task and 6 ms for a scribbling task. Using the same apparatus, Annett et al. studied writing and drawing tasks, and found that users could perceive latencies down to 50 ms [3]. The combined findings of Ng et al. and Annett et al. suggest that latency perception is dependent on the task, and can still be perceived well below the latencies of 55 to 200 ms provided by currently available digital pens.

These direct and indirect studies educate us on specific tasks or form-factors, but there are no single comparisons between direct and indirect systems. Furthermore indirect studies have not taken advantage of the recently available low latency prototypes [12,19], which might explain why the results of direct input studies that do take advantage of this hardware are significantly lower. We therefore set out to conduct a series of experiments that leveraged low latency hardware to directly compare the perception of latency on both direct and indirect form-factors.

Measuring Latency Perception: Just Noticeable Difference

A Just Noticeable Difference (JND) is the minimum difference in a pair of stimuli that is detectable by a person, and can be measured for any perceptual stimuli (e.g., light, pressure, sound). A trial in a JND experiment [13] presents a pair of different stimuli (*A* and *B*) to a participant and asks them to identify which is faster (or brighter, louder, etc.). One stimulus, the *reference*, is held constant throughout the experiment; the second stimulus, the *probe*, is varied in each trial. The probe begins far from the reference and is moved closer to the reference whenever a participant is able to correctly distinguish them, and further from the reference when they cannot. The series of probe values is termed a *staircase*, due to its series of up and down movements. A *reversal* in the direction of the staircase occurs when a correct answer is given after an incorrect response, or vice versa. After a series of trials, the probe will converge at the point where the participant is just able to distinguish it from the reference and will oscillate indefinitely above and below this point (given a sufficiently patient participant). This threshold is termed the *JND threshold*.

EXPERIMENT 1: PERCEPTION OF LATENCY IN DIRECT AND INDIRECT POINTING TASKS

In this first investigation, we aimed to clarify and extend previous efforts on measuring the minimal perceivable latency for common touch-based interactions. We conducted a pair of JND studies in which participants performed either tapping or dragging tasks in both direct and indirect form-factors. We selected dragging and tapping tasks because they represent the basic input primitives used in most user interfaces. Our goals with this experiment were two-fold. First, we aimed to gather data for finger-based indirect (i.e., touchpad) interaction, which had previously been unexplored. Second, we aimed to gather a unified set of data for both direct and indirect form-factors with an identical set of hardware and experimental conditions in order to enable an apples-to-apples comparison.

Hypotheses

Our hypotheses center on the belief that input task and form-factor play major roles in peoples' ability to perceive input latency and that these differences explain the variety of reported JNDs in the literature.

Dragging provides the most visible manifestation of latency, since an on-screen object will begin to trail behind a user's finger as the latency increases. We hypothesize that this physical distance between a finger and the graphical cursor is more easily perceived than the purely temporal difference between a tap and its graphical response.

> *H1: Users will perceive a lower JND threshold when dragging than when tapping.*

With direct input, the finger and graphical response are co-located, allowing the user to perceive both through the visual channel. We hypothesize that the visual difference between input and response in the direct form-factor is more easily perceivable than the difference between the kinesthetic touch and visual response in indirect input.

> *H2: Users will perceive a lower JND threshold with direct input than with indirect input.*

Apparatus

We used a high-speed Fast Multi-Touch (FMT) sensor and projector similar to that described in Leigh et al. [12] and illustrated in Figure 1.The system provides a 15 cm × 20 cm capacitive touch sensor, which would have provided too much freedom of movement for our participants; since we were interested in the JND thresholds for dragging and tapping input primitives, unconstrained movements over the entire touch sensor would have provided an unwanted confound. We therefore placed a cardboard mask on the touch surface to constrain the participant's movements to the appropriate input method, thereby ensuring a one-dimensional movement during a dragging trial and preventing any dragging movement during a tapping trial. Dragging trials used a 14 cm × 3 cm rectangular slot while tapping trials used a 2 cm square. A custom-built controller box with illuminated mechanical push buttons informed the participant whether stimulus *A* or *B* was currently visible and allowed the participant to provide input (e.g., switching between the two stimuli and entering their responses). The box was powered by an Arduino microcontroller and connected to the control laptop via USB. Simple audio feedback (e.g., beeps and tones) was provided to indicate the start of each trial, the end of each block, and whenever the user made an invalid selection on the control box.

In its normal configuration, the FMT hardware is a direct manipulation system; the touch surface is parallel to the desk, and the high-speed projector is mounted on a support arm above the surface, top-projecting an image onto the touch sensor. To convert the system to an indirect form-factor, we designed a removable bracket containing a front-surface mirror mounted on a 45° angle. When the mirror assembly was in place, the path of the projector's beam was redirected so that it appeared on a reflective screen mounted on the wall perpendicular to the touch surface, thereby simulating an indirect setup akin to a laptop touchpad and screen. The mirror assembly could be quickly inserted or removed, facilitating a rapid transition between form-factors. The positions of the mirror and screen were adjusted so that the images drawn by the projector were of the same apparent size, regardless of whether they were projected on the touch surface or on the wall. To maintain the same reflectivity and apparent brightness, the screen was covered in the same contact paper used in the top-most layer of the touch sensor. In both cases, the visual feedback provided after a touch was a solid white 2 cm square.

The FMT hardware is capable of running at latencies as low as 0.29 ms as well as at a wide range of higher latencies. However, it cannot run at any arbitrary latency; the possible latencies are discretely quantized based on two parameters: the sample interval and the queue length. The sample interval can be configured from 0.29 ms to 25.07 ms, in step sizes of 0.098 ms. The queue length is the number of samples buffered prior to processing, and ranges from 1 to 255. The overall latency of the system is the product of these two parameters. For example, a sampling interval of 0.98 ms and a queue length of 10 results in a latency of $0.98 \times 10 = 9.8$ ms. For this study, we held the sampling interval constant at 0.98 ms (the closest value to 1 ms) and varied the queue length to alter the latency; this yielded a possible range of latencies of 0.98 ms to 250.67 ms, with a step size of 0.98 ms. It should be noted that while these non-integer latency values do not impact the analysis of the data, they do result in atypical values for some of the experimental parameters (e.g., the staircase step size is 7.86 ms, a value which corresponds to a queue length of 8). At the start of each trial, parameters were sent to the FMT hardware by a MacBook Pro laptop running custom control software connected over a dedicated Ethernet connection. We empirically validated a variety of latencies using an oscilloscope triggered by vibration and photo sensors, as well as with a high-speed camera.

Procedure

After participants were briefed on the study, they completed a short warm-up session. Participants then began the main experiment consisting of four blocks of trials. For each trial, a participant was first presented with latency *A* and asked to tap or drag; all interactions in a single session used the same input technique. After trying latency *A*, the participant used the control box to switch to latency *B* and performed the interaction again. At this point, they could either switch back to see latency *A* a second time or indicate their decision as to which latency they thought was shorter[1]. If they elected to see latency *A* a second time, they were then required to make a decision and could not switch back to latency *B*. Each trial was a forced choice. When participants could not distinguish the two stimuli, they were instructed to make their best guess [23]. After they entered their decision, the system would beep and move to the next trial. Halfway through the session, after the first two blocks, the mirror assembly was inserted or removed to change the form-factor of the device and the final two blocks of trials were completed with the second form-factor. At the end of the session, participants were quickly debriefed about the latency detection techniques they had used.

For tapping tasks, participants were instructed to use their right index finger to press and release inside the target region. Participants could tap as many times as they wanted and could hold the tap for any length of time. However, extremely rapid tapping (i.e., oscillating the finger up and down as quickly as possible) was not permitted.

For dragging tasks, participants were instructed to use their right index finger to press down on the left side of the target, move their finger to the right side of the target, and then back to the left. They could drag as many times as they wanted and at any speed. The dragging cardboard mask was sized so that the drag would be a predominantly one-dimensional left/right movement on the x-axis, with little or no y-axis movement towards or away from the user.

Sessions were designed to be completed in 90 to 120 minutes. Participants were permitted breaks at any point in the experiment, either between blocks or within them.

[1] The term "shorter latency" was not used during the experiment since it was felt that it might be needlessly confusing to non-computer scientists. Instead, participants were instructed to choose the latency that was "faster", "more responsive", and "most closely following your finger".

Participants

A total of 24 sessions were run across both form-factors (12 drag and 12 tap). The sessions were performed by 14 right-handed participants (6 male) recruited from the broader university community, with a mean age of 28 (sd = 5.4). All had experience with consumer touchscreen devices. Participants were offered the opportunity to perform either one or two sessions (one of each form-factor), separated by at least two hours to account for learnability biases and fatigue effects. 10 participants chose to participate in both sessions. Participants were compensated $20 for each session they completed.

Design

Each session consisted entirely of either a tapping or dragging JND study. Participants were randomly assigned to one type of session; those that participated in two sessions were randomly assigned the first session, and were given the other input technique in their second session.

The latencies used in each trial were generated using an adaptive staircase algorithm. The reference was held constant at 0.98 ms (i.e., a FMT queue length of 1) throughout the experiment. The second latency, the probe, was varied according to Kaernbach's simple weighted up-down method [11]. A base step size of 7.86 ms (i.e., queue length of 8) was used, which was halved upon each of the first three reversals until it reached 0.98 ms. Decreases in the probe were reduced by the base step size while increases were increased by three times the base step size.

A total of eight staircases were run for each participant, arranged in four blocks that each contained a pair of interleaved staircases. Two blocks were run using one form-factor, followed by two blocks using the other form-factor. The ordering of the form-factors was counterbalanced. Within a block, trials from the two staircases were interleaved to prevent the participant from being able to (consciously or subconsciously) identify a pattern in the trials. For interleaved tapping staircases, one began at 117.96 ms and the other began at 58.98 ms; dragging staircases began at 98.30 ms and 39.32 ms. Within each trial, the order of presentation of the two latencies was randomized.

The length of each staircase was not fixed in advance; a block continued until both staircases reached 10 reversals, yielding a 75% confidence threshold for the JND threshold [13]. In most cases one staircase reached 10 reversals before the other one since the order of correct/incorrect responses in each staircase is extremely unlikely to be synchronized. When this occurred, trials from the completed staircase continued to be alternated with the unfinished staircase in order to maintain the interleaving. This meant that a block would usually yield more than 10 reversals for the staircases that finished first. When analyzing the staircases, we ignored reversals past the 10th.

The overall design of the experiment was:

12 sessions ×
2 input techniques (*tapping*/*dragging*) ×
2 form-factors (*direct*/*indirect*) ×
2 repetitions =
96 staircases

As indicated above, the number of trials in a staircase depended on the pattern of responses. The average length across all staircases was 89.1, with a total of 8,556 comparisons of latency pairs in the raw dataset.

Results

To analyze our results, we performed a repeated-measures ANOVA using *Input Task* and *Form-Factor* as between-participant independent variables. Our dependent variable was *JND*, which was the average of the thresholds found from the two staircases in a block. To justify the use of a between-participants analysis, we first checked for asymmetrical transfer effects by performing a within-participant analysis to look for effects from *Form-Factor* ordering (direct-then-indirect vs. indirect-then-direct) or *Repetition*. Finding none, we were able to proceed with the between-participant design.

Our results confirmed both hypotheses. We found a significant main effect from *Input Task* ($F_{1,22} = 55.79$, $p < 0.001$, $\eta^2 = 1.00$), with mean latency *JND*s of 33 ms and 82 ms for *Dragging* and *Tapping* tasks respectively. There was also a significant main effect from *Form-Factor* ($F_{1,22} = 48.43$, $p < 0.001$, $\eta^2 = 1.00$), with mean latency *JND*s of 40 ms and 75 ms for *Direct* and *Indirect* input respectively. No other effects or interactions were detected.

To dig deeper into the results, we computed two additional ANOVAs for the *Tapping* and *Dragging* tasks independently. For *Dragging*, *Form-Factor* had a significant main effect on *JND* ($F_{1,11} = 77.11$, $p < 0.001$, $\eta^2 = 1.00$) with mean *JND*s of 11 ms and 55 ms for *Direct* and *Indirect* input. Equivalent results were also found for *Tapping*, where *Form-Factor* also had a significant main effect on *JND* ($F_{1,11} = 9.44$, $p = 0.011$, $\eta^2 = 0.80$), with mean *JND* values of 69 ms and 96 ms for *Direct* and *Indirect* input. Combined *JND*s (counting both tasks for a given form-factor) were 40 ms (*Direct*) and 75 ms (*Indirect*). The mean latency *JND*s are shown in Figure 2.

Discussion

Our results confirm our hypotheses surrounding the importance of form-factor and input task on the limits of latency perception. It is clear that people are much better overall at noticing latency with direct input touch devices than with indirect ones, suggesting that the threshold that system designers must reach in order to provide perceptually “latency-free” experiences is higher for touchpads than for touchscreens. Similarly, users perceive latency at a much finer level when dragging than when tapping. Indeed, with direct input, dragging provides the most visible manifestation of latency, since an on-screen object will

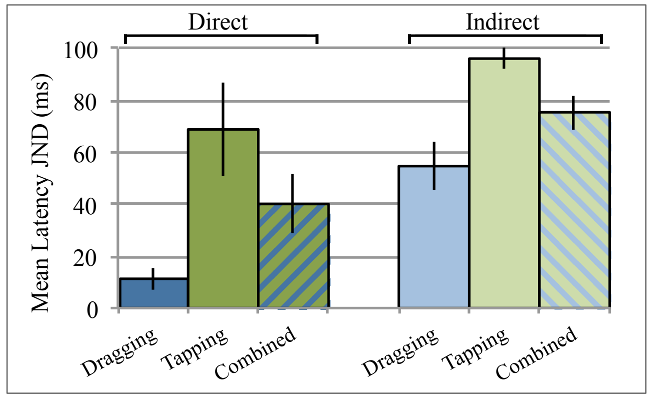

Figure 2. Mean latency JND thresholds for direct and indirect form-factors. Dragging, tapping, and combined (counting trials from both techniques) are shown in blues, greens, and striped bars. Error bars show 95% confidence intervals.

begin to trail behind a user’s finger as the latency increases, creating a readily perceivable spatial distance between the two. This relationship between input task and latency perception was discussed by Jota et al. [8] in their own comparison to Ng et al. [19]; however, our results indicate that this relationship extends to indirect form-factors as well. Recent UI trends towards more fluid dragging actions are therefore likely to amplify the perceived latency in both form-factors and encourage system designers to further improve the latency of their devices.

As a whole, these results bring some clarity to a disagreement in the literature over latency JNDs since they suggest that form-factor and input task must be considered. Ng et al. [19], who report a JND of 6.04 ms, argue that the textbook threshold of 100 ms (a value based on Miller’s work [16] and guidelines for audio in the telecommunications industry [4], among other sources) is incorrect and an artifact of the limited performance of input devices and measurement techniques available at the time of its origin. Our results suggest that both are correct, as Ng et al. report on a direct input dragging task and most others report on a variety of indirect input tapping tasks (menu selection, key typing, path drawing, etc.). Indeed our reported mean JNDs of 11 ms, 69 ms, and 96 ms for direct dragging, direct tapping, and indirect tapping correspond very closely to the reported JNDs of 6.04 ms (Ng et al., direct dragging), 64 ms (Jota et al. [8], direct tapping), and 100 ms (textbook guideline) – with all of these previously reported values falling well within the 95% confidence intervals of our estimated means.

EXPERIMENT 2: UNDERSTANDING INCREMENTAL IMPROVEMENTS ABOVE THE LOWER BOUND FOR DRAGGING AND TAPPING TASKS

The results from Experiment 1 inform system designers as to what latency targets they must meet in order to provide a perceptually “latency-free” UI. As stated in the introduction, the vast majority of commercial devices fall well above these limits of perception; thus, reaching this goal is likely to be a long and laborious process.

In the meantime, it is desirable to better understand what incremental improvements to latency would mean in terms of users' perception of responsiveness. Early efforts investigating latency perception sought to find the lower bounds of perception, and although Gaussian models of perception can provide predictions regarding performance at interim latency improvements [7], we were interested in concretely mapping the relationship between touch input latency and user perception at levels typical of current consumer-level hardware.

Why might an understanding of this relationship be valuable? Consider an OEM whose touchscreen device operates with 100 ms of latency. Would reducing this latency to 90 ms be noticeable by their users? How about a reduction to 80 ms? If that 20 ms improvement was applied to a device that operates at 50 ms, would that be noticed? Perhaps there would be no noticeable difference until a device passed some intermediate threshold. Similarly, an OEM wishing to produce a lower-cost device might consider components that would increase the latency, and may wish to know if and how much this increase would be perceived by their customers. When considering such questions, one understands the value of examining the perception of input latency improvements above the lower bound. Experiment 2 aims to begin this valuable investigation for both tapping and dragging. Not knowing what we would find, it was difficult to generate specific hypotheses; as such, we proceeded with an exploratory study to help enhance our understanding of the space.

Display Refresh Rate

One major contributor to system latency in current touch devices is display refresh rate. No matter how quickly new graphical information is ready, it cannot be displayed as soon as it is available as it must wait to appear until the display can redraw its image. At the time of this writing, 60 Hz displays are still the norm, meaning that a new frame is drawn every 1000 ms / 60 = 16.67 ms. This refresh rate has the perverse effect of quantizing and possibly masking improvements to latency that fall out of sync with this 16.67 ms heartbeat. As a concrete example, consider a 60 Hz system with a latency of 32 ms. Such a device will be able to update its UI within two frames. If this latency was reduced to 16 ms, then it would be able to update within one frame, resulting in a displayable difference. However, if the latency was reduced from 32 ms to 25 ms, the system would still require two frames before a UI update could occur, meaning that the reduction down to 25 ms will have had no visible impact. This observation governed many of the magnitude choices used in our experiment as we gravitated to base latencies and latency improvements that were multiples of a 60 Hz display.

Apparatus and Participants

We used the apparatus described in Experiment 1. For this experiment, we recruited 15 right-handed participants (7 male) from the broader university community, with a mean age of 25 (sd = 4.3). Nine of our participants took part in both the dragging and tapping sessions, for a total of 24 sessions (12 tapping and 12 dragging). Participants were compensated $20 per session.

Design and Procedure

As in Experiment 1, each session consisted of a series of blocks of A/B trials in which the participant was asked to identify the faster latency. Each session consisted entirely of either tapping or dragging, and contained a total of 8 blocks; the first 4 consisted of one form-factor, and the last 4 of the other. As with Experiment 1 the order of the form-factors was counterbalanced. Each pair of latencies consisted of a *baseline* latency, and that *baseline* minus a *difference*. All blocks contained the same set of latencies pairs; the order of the pairs was randomized, and the A/B position of the individual latencies within a pair were counter-balanced so that each latency appeared twice in the A (first) position and twice in the B (second) position.

To determine our set of baselines, we considered the common case of a 60 Hz display, which displays a new frame every 16.67 ms, as discussed above. We therefore selected baselines that corresponded to updates occurring every 1 to 10 frames: 16.7, 33.3, 50.0, 66.7, 83.3, 100.0, 116.7, 133.3, 150.0, and 167.7 ms. For completeness, we also added a baseline of 8.3 ms (0.5 frames), for a total of 11 baselines. We then selected differences that corresponded to 0.5, 1, 2, 3, 4, and 5 frames of improvement: 8.3, 16.7, 33.3, 50.0, 66.7, and 83.3 ms. Each difference was subtracted from each baseline. Pairs that resulted in a negative latency (e.g., 8.3 ms − 16.7 ms) were excluded, and differences that resulted in 0 ms (e.g., 8.3 ms − 8.3 ms) were rounded up to 1 ms, the minimum latency that the experimental configuration of the system could produce. This resulted in a total of 51 pairs, which constituted the contents of one block of the experiment. All latencies were then converted to the nearest latency that could be reproduced on the FMT hardware, as described in the Apparatus section of Experiment 1 (e.g., 100 ms was converted to 100.27 ms); this conversion resulted in differences of less than 0.5 ms (0.3%). The overall design of the experiment was:

12 sessions ×
2 input techniques (*tapping*/*dragging*) ×
2 form-factors (*direct*/*indirect*) ×
4 repetitions ×
51 valid comparisons/repetition =
9,792 trials.

Results

As in Experiment 1, we first checked for asymmetrical learning effects by looking for an ordering effect from *Form-Factor*. Finding none, we proceeded with the analysis presented in the following sections. While not included in the same statistical model, a comparison between *Dragging* and *Tapping* input reveals a large difference in latency perception. Overall, participants correctly identified improvements to latency 81.6% of the time when *Dragging*

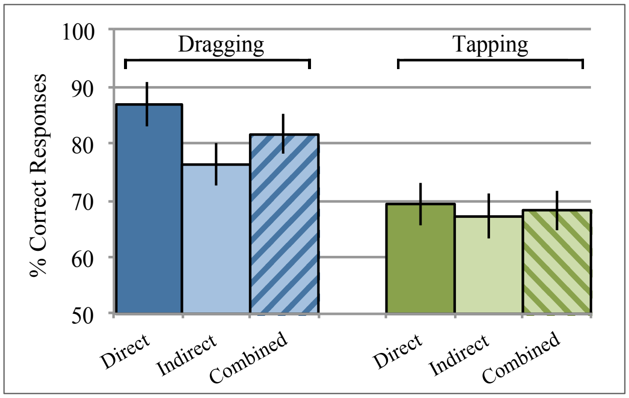

Figure 3. Mean percent correct for dragging and tapping. Direct, indirect, and combined (counting trials from both form-factors) are shown in darker, lighter, and striped bars. As a group, participants are better at recognizing improvements to latency when dragging than when tapping. Error bars show 95% confidence intervals.

and only 68.2% of the time when *Tapping*. The lower bound of the 95% confidence interval for both means fall well above the 50% chance value one would expect if users were guessing. Figure 3 shows the mean percentage of correct trials for each input technique for both direct and indirect form-factors as well as the overall percentage combining all trials of both form-factors.

Dragging Analysis

We conducted a repeated-measures ANOVA using *Form-Factor* (*Direct/Indirect*), *Latency Improvement*, and *Base Latency* as independent variables, and *Percent Correct* as a dependent variable.

As expected, *Form-Factor* had a significant main effect on *Percent Correct* ($F_{1,11} = 47.56$, $p < 0.001$, $\eta^2 = 1.00$) with participants correctly identifying differences in direct input latency 86.9% of the time, compared to 76.3% of the time with indirect input. As was the case for the direct dragging JND threshold, this combination of technique and form-factor provides the most visible manifestation of latency. In addition, *Latency Improvement* also had a significant main effect ($F_{1,11} = 118.31$, $p < 0.001$, $\eta^2 = 1.00$), with larger *Latency Improvements* being easier to notice. A post-hoc pairwise comparison shows a significant difference between most pairs of *Latency Improvements*. Figure 4 shows the mean *Percent Correct* for each *Latency Improvement*, broken down by form-factor.

Overall, there was no significant main effect on *Percent Correct* from *Base Latency* ($F_{10,110} = 0.85$, $p = 0.58$, $\eta^2 = 0.43$); however, there was a significant interaction between *Base Latency* and *Form-Factor* ($F_{10,110} = 3.76$, $p < 0.001$, $\eta^2 = 0.99$). In general, participants' ability to identify improvements in indirect latency is consistent across *Base Latencies*, but is more accurate in the lower *Base Latencies* when working in the *Direct Input Form-Factor*. This may be due to the inherent perceptual advantage associated with a large percentage difference between stimuli (i.e., an 8.3 ms decrease from 16.7 ms is a 50% decrease, while an 8.3 ms decrease from 166.7 ms is only 5%), and/or the fact that a decrease from a smaller baseline is more likely to result in a perceptually "zero latency" stimulus once the post-decrease value crosses the JND thresholds established in Experiment 1. In a trial where one of the two stimuli appears to be instantaneous, the task of discriminating the pair becomes easier, since the task is reduced to the simpler question of identifying the single stimulus that had any visible latency at all.

Figure 5 shows the mean *Percent Correct* for each *Base Latency / Form-Factor* combination. There were no other significant effects or interactions found in the *Dragging* activity.

While the effects of *Activity*, *Base Latency*, *Latency Improvement*, *Form-Factor* and their interactions are an interesting and useful topic for study, our primary motivation for this experiment was to work toward answering the question "Can people recognize improvements to latency *across the latency spectrum*?"

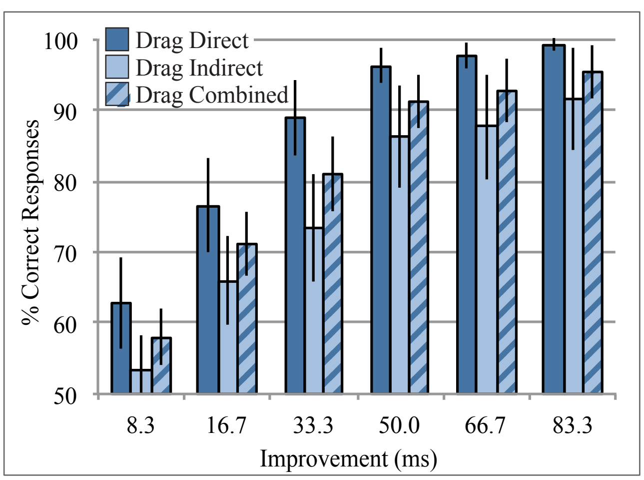

Figure 4. Mean percent correct for each dragging latency improvement. Error bars show 95% confidence intervals.

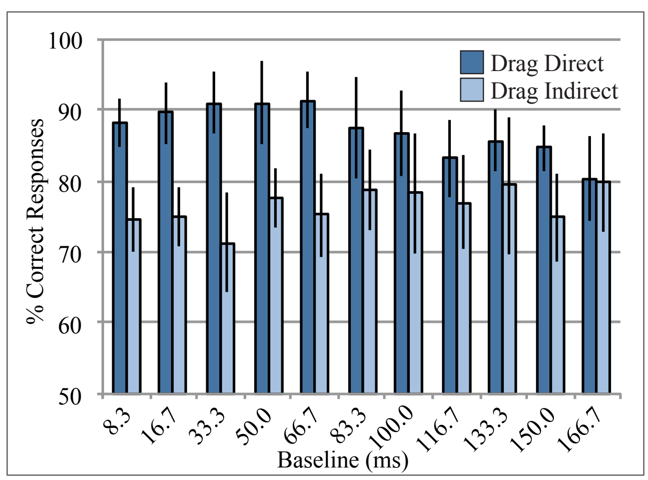

Figure 5. Mean percent correct for each dragging base latency. Error bars show 95% confidence intervals.

Previous efforts (in this paper and others) have investigated the limits of human perception in respect to near-zero latency, but we are equally interested in whether or not people can perceive improvements to latency away from this lower bound.

Table 1 shows the percentage of trials in which our participants (as a group) correctly identified the lower latency dragging interaction. These tables show each *Base Latency* / *Latency Improvement* combination. Cells marked with an asterisk have true means that are within a 95% confidence interval of the 50% chance threshold, and are therefore not statistically distinguishable from chance (i.e., combinations where the reduction in latency was not perceived), and are in the clear minority. Colored regions of the Tables are areas that are discussed in more detail below.

Dragging Discussion

Overall, the results from our exploration into the perception of latency improvements when dragging shows that there are not only clear effects from *Base Latency* and *Latency Improvement*, but also significant room for perceivable improvement to latency without eliminating it altogether. Considering Table 1, for the *Direct Dragging* condition, any improvement in latency of 1 frame/sec (fps) or more is easily observable by our participants as a whole (uncolored area). Furthermore, even a small improvement of 0.5 fps is observable by a significant fraction of our participants for base latencies at or under 3 fps (orange area). For systems with higher base latencies, this small improvement did not yield observable differences (blue area). For *Indirect Dragging* input, there are clearly observable improvements to be had for latency improvements of 2 or more fps (uncolored area), with less value for smaller improvements of 0.5 or 1 fps (yellow area). These trends seem relatively stable for base latencies across the tested spectrum.

Tapping Analysis

We ran another repeated-measures ANOVA using *Form-Factor* (*Direct/Indirect Input*), *Latency Improvement*, and *Base Latency* as independent variables, and *Percent Correct* as a dependent variable.

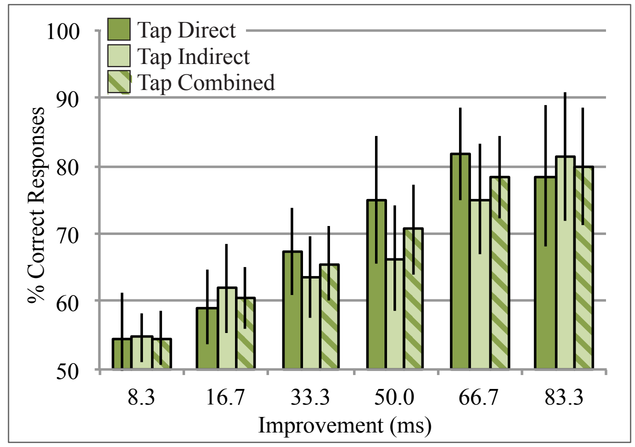

Figure 6. Mean percent correct for each tapping latency improvement. Error bars show 95% confidence intervals.

Unlike with *Dragging*, we did not find a significant main effect for *Form-Factor* on *Percent Correct* ($F_{1,11} = 2.18$, $p = 0.17$, $\eta^2 = 0.27$), with participants as a whole correctly identifying the lower latency in 69.3% and 67.1% of *Direct Input* and *Indirect Input* trials respectively.

Again, as expected, *Latency Improvement* also had a significant main effect ($F_{1,11} = 19.55$, $p = 0.001$, $\eta^2 = 0.98$), with larger *Latency Improvements* being easier to notice in the *Tapping* task. A post-hoc pairwise comparison between means shows a significant difference between all pairs of *Latency Improvement*. Figure 6 shows the mean *Percent Correct* for each *Latency Improvement*.

Base Latency had a significant main effect on *Percent Correct* ($F_{1,11} = 8.85$, $p = 0.01$, $\eta^2 = 0.77$), with improvements to higher *Base Latencies* being correctly identified more often than those to lower *Base Latencies*. Figure 7 shows the mean *Percent Correct* for each *Base Latency* in the *Tapping* task. There were no significant interactions found in the *Tapping* analysis.

Dragging Direct	Baseline Latency (ms)										
Improvement (ms)	8.3	16.7	33.3	50.0	66.7	83.3	100.0	116.7	133.3	150.0	166.7
8.3	60%	69%	69%	73%	67%	60% *	75%	54% *	60% *	54% *	50% *
16.7	-	77%	85%	79%	85%	75%	69%	63% *	75%	83%	73%
33.3	-	-	90%	92%	94%	94%	94%	90%	85%	81%	79%
50.0	-	-	-	100%	100%	96%	90%	94%	92%	98%	90%
66.7	-	-	-	-	100%	98%	96%	98%	100%	92%	92%
83.3	-	-	-	-	-	100%	96%	100%	100%	98%	98%

Dragging Indirect	Baseline Latency (ms)										
Improvement (ms)	8.3	16.7	33.3	50.0	66.7	83.3	100.0	116.7	133.3	150.0	166.7
8.3	54% *	56% *	42% *	54% *	50% *	56% *	54% *	48% *	54% *	65%	52% *
16.7	-	60% *	54% *	69%	75%	75%	69% *	69% *	67% *	54% *	73%
33.3	-	-	69%	79%	58% *	79%	73%	79%	81%	69%	81%
50.0	-	-	-	83%	88%	85%	88%	85%	92%	88%	90%
66.7	-	-	-	-	88%	83%	94%	90%	92%	81%	88%
83.3	-	-	-	-	-	92%	92%	90%	90%	92%	94%

Table 1. Mean percent correct for dragging trials for each combination of base latency / latency improvement. Cells marked with an asterisk are not significantly different from chance (50%). Colored regions are areas of interest that are discussed in the text. 95% confidence intervals range from 5% to 21%.

As with *Dragging* interactions, we are keenly interested in whether or not small improvements to latency are perceivable to participants across the latency spectrum. Table 2 shows the percentage of trials in which our participants (as a group) correctly identified the lower latency tapping interaction. These tables show each *Base Latency / Latency Improvement* combination; as with Table 1, cells marked with an asterisk are statistically indistinguishable from chance. While there are clear effects from *Base Latency* and *Latency Improvement*, these tables are included to suggest that there is a great deal of room for perceivable improvement to latency without eliminating latency entirely.

Tapping Discussion

Considering Table 2, it appears that for systems with base latencies at or below 33.3 ms there is little room for improvement in latency perception in either form-factor (red areas). When considering systems that have a higher base latency, improvements of 2 or more fps seem to have an observable difference (uncolored area) for all base latencies tested. Smaller improvements of 0.5 or 1 fps were generally not observable to our participants (purple area).

CONCLUSION

In this paper, we have examined user's perception of latency for both dragging and tapping tasks under both direct and indirect form-factors. A set of JND studies indicated that the detectable thresholds for dragging (direct: 11 ms, indirect: 55 ms) are lower than for tapping (direct: 69 ms, indirect: 96 ms), and that direct touch systems are more susceptible to noticeable latency than their indirect counterparts. A second set of studies demonstrated that improvements in latency as small as 8.3 ms are noticeable from a wide range of baseline latencies, particularly when dragging.

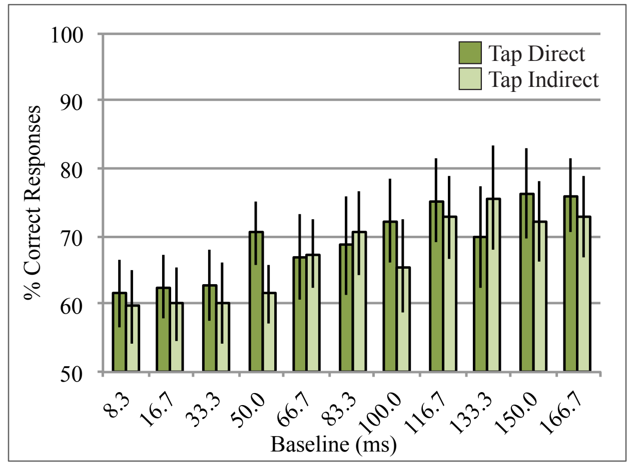

Figure 7. Mean percent correct for each tapping base latency. Error bars show 95% confidence intervals.

While the end goal of a zero-latency system will of course require a significant engineering effort, our results provide clear guidance to system designers that interim steps along that path are in fact worthwhile. The latency improvements that result from the removal of just a single 60 Hz frame (i.e., a decrease of 16.7 ms in latency) are perceptibly noticeable under many circumstances.

While we are happy to report some clarity on users' ability to *perceive* latency improvements (not only at the lower bound, but at higher levels as well), further effort is required. Next steps will include understanding the *desirability* of such improvements to a user, as well as any benefits to *performance* in common touch tasks, such as tapping, docking, crossing, and scrolling. We have begun investigating these follow-on efforts as we continue to strive to understand and provide zero-latency touch experiences.

Tapping Direct	Baseline Latency (ms)										
Improvement (ms)	8.3	16.7	33.3	50.0	66.7	83.3	100.0	116.7	133.3	150.0	166.7
8.3	46% *	52% *	56% *	56% *	48% *	52% *	50% *	73%	56% *	58% *	52% *
16.7	-	50% *	48% *	69%	58% *	60% *	65%	60% *	60% *	58% *	71%
33.3	-	-	52% *	77%	73%	71%	63% *	75%	69%	81%	77%
50.0	-	-	-	69%	71%	73%	79%	79%	77%	92%	79%
66.7	-	-	-	-	79%	83%	85%	77%	79%	88%	90%
83.3	-	-	-	-	-	73%	92%	88%	77%	81%	88%

Tapping Indirect	Baseline Latency (ms)										
Improvement (ms)	8.3	16.7	33.3	50.0	66.7	83.3	100.0	116.7	133.3	150.0	166.7
8.3	50% *	52% *	50% *	46% *	58% *	56% *	44% *	56% *	67%	60% *	63%
16.7	-	58% *	60% *	58% *	63% *	69%	54% *	69%	63% *	69%	60% *
33.3	-	-	48% *	63%	67%	69%	67%	77%	75%	69%	71%
50.0	-	-	-	56% *	71%	79%	69%	71%	73%	69%	73%
66.7	-	-	-	-	67%	71%	77%	90%	92%	79%	83%
83.3	-	-	-	-	-	79%	83%	75%	85%	88%	88%

Table 2. Mean percent correct for tapping trials for each combination of base latency and latency improvement. Cells marked with an asterisk are not significantly different from chance (50%). Colored regions are areas of interest that are discussed in the text. 95% confidence intervals range from 7% to 19%.

ACKNOWLEDGMENTS

We would like to thank all of our colleagues at DGP and at Tactual Labs, as well as Katie Barker. Special thanks to Haijun Xia for his assistance with the videos.

REFERENCES

1. Allison, R.S., Harris, L.R., Jenkin, M., Jasiobedzka, U., and Zacher, J.E. (2001). Tolerance of Temporal Delay in Virtual Environments. In *Proc. IEEE VR '01*, 247–254.
2. Anderson, G., Doherty, R., and Ganapathy, S. (2011). User Perception of Touch Screen Latency. In *Proc. Design, User Experience, and Usability (DUXU) '11*, 195–202.
3. Annett, M., Ng, A., Dietz, P., Bischof, W.F., and Gupta, A. (2014). How Low Should We Go? Understanding the Perception of Latency While Inking. In *Proc. GI '14*,167–174.
4. Cheshire, S. (1996). *Latency and the Quest for Interactivity*. White paper for the Synchronous Person-to-Person Interactive Computing Env Meeting, Nov. 1996.
5. Ellis, S.R., Bréant, F., Manges, B., Jacoby, R., and Adelstein, B.D. (1997). Factors Influencing Operator Interaction with Virtual Objects Viewed via Head-Mounted See-Through Displays: Viewing Conditions and Rendering Latency. In *Proc. IEEE VR '97*, 138–145.
6. Ellis, S.R., Young, M.J., Adelstein, B.D., and Ehrlich, S.M. (1999). Discrimination of Changes of Latency During Voluntary Hand Movements of Virtual Objects. In *Proc. Human Factors and Ergonomics Society '99*, 1182–1186.
7. Harris, L., Harrar, V., Jaekl, P., and Kopinska, A. (2010). Mechanisms of Simultaneity Constancy. In Nijhawan, R., and Khurana, B. (Eds.). *Space and Time in Perception and Action*. Cambridge: Cambridge University Press, 232–253.
8. Jota, R., Ng, A., Dietz, P., and Wigdor, D. (2013). How Fast is Fast Enough? A Study of the Effects of Latency in Direct-Touch Pointing Tasks. In *Proc. CHI '13*, 2291–2300.
9. Kaaresoja, T., and Brewster, S. (2010). Feedback is… Late: Measuring Multimodal Delays in Mobile Device Touchscreen Interaction. In *Proc. ICMI '10*, Article 2, 8 pages.
10. Kaaresoja, T., Brewster, S., and Lantz, V. (2014). Towards the Temporally Perfect Virtual Button: Touch-Feedback Simultaneity and Perceived Quality in Mobile Touchscreen Press Interactions. *ACM Trans. Appl. Percept*. 11, 2, Article 9 (Jun. 2014), 25 pages.
11. Kaernbach, C. (1991). Simple Adaptive Testing with the Weighted Up-Down Method. *Perception & Psychophysics* 49, 3 (Mar. 1991), 227–229.
12. Leigh, D., Forlines, C., Jota, R., Sanders, S., and Wigdor, D. (2014). High-Rate, Low-Latency Multi-Touch Sensing with Simultaneous Orthogonal Multiplexing. In *Proc. UIST '14*, 355–364.
13. Levitt, H. (1971). Transformed up-down methods in psychoacoustics. *J. Acoustical Soc. of Am*. 49, 2 (Feb. 1971), 467–477.
14. MacKenzie, I.S., and Ware, C. (1993). Lag as a Determinant of Human Performance in Interactive Systems. In *Proc. CHI '93*, 488–493.
15. Meehan, M., Razzaque, S., Whitton, M.C., and Brooks, F.P. (2003). Effect of Latency on Presence in Stressful Virtual Environments. In *Proc. IEEE VR '03*, 141–138.
16. Miller, R.B. (1968). Response Time in Man-Computer Conversational Transactions. In *Proc. AFIPS '68*, 267–277.
17. Nelson, W.T., Roe, M.M., Bolia, R.S., and Morley, R.M. (1998). Assessing Simulator Sickness in a See-Through HMD: Effects of Time Delay, Time on Task, and Task Complexity. In *Proc. IMAGE '00*.
18. Ng, A., Annett, M., Dietz, P., Gupta, A., and Bischof, W.F. (2014). In the Blink of an Eye: Investigating Latency Perception During Stylus Interaction. In *Proc. CHI '14*, 1103–1112.
19. Ng, A., Lepinski, J., Wigdor, D., Sanders, S., and Dietz, P. (2012). Designing for Low-Latency Direct-Touch Input. In *Proc. UIST '12*, 453–464.
20. Pavlovych, A., and Gutwin C. (2012). Assessing Target Acquisition and Tracking Performance for Moving Targets in the Presence of Latency and Jitter. In *Proc. GI '12*, 109–116.
21. Pavlovych, A., and Stürzlinger, W. (2011). Target Following Performance in the Presence of Latency, Jitter, and Signal Dropouts. In *Proc. GI '11*, 33–40.
22. Pavlovych, A., and Stürzlinger, W. (2009). The Tradeoff between Spatial Jitter and Latency in Pointing Tasks. In *Proc. ACM EICS '09*, 187–196.
23. Savage, W.C. (1970). *The Measurement of Sensation: A Critique of Perceptual Psychophysics*. Berkley: UC Press.
24. So, R.H.Y., and Chung, G.K.M. (2005). Sensory Motor Responses in Virtual Environments: Studying the Effects of Image Latencies for Target-directed Hand Movement. In *Proc. IEEE-EMBS '05*, 5006–5008.
25. Steed, A. (2008). A Simple Method for Estimating the Latency of Interactive, Real-Time Graphics Simulations. In *Proc. ACM VRST '08*, 123–129.
26. Teather, R., Pavlovych, A., Stürzlinger, W., and MacKenzie, I.S. (2009). Effects of Tracking Technology, Latency, and Spatial Jitter on Object Movement. In *Proc. 3DUI '09*, 43–50.
27. Vogels, I.M.L.C. (2004). Detection of Temporal Delays in Visual-Haptic Interfaces. *Hum. Fact.* 46, 1 (Spr. 2004), 118–134.
28. Ware, C., and Balakrishnan, R. (1994). Reaching for Objects in VR Displays: Lag and Frame Rate. *ACM Trans. Comput.-Hum. Interact*. 1, 4 (Dec. 1994), 331–356.