

Do We Need a Faster Mouse? Empirical Evaluation of Asynchronicity-Induced Jitter

Auejin Ham
KAIST
Daejeon, Republic of Korea
hamaj@kaist.ac.kr

Junsu Lim
DGIST
Daegu, Republic of Korea
junsu@dgist.ac.kr

Sunjun Kim*
DGIST
Daegu, Republic of Korea
sunjun_kim@dgist.ac.kr

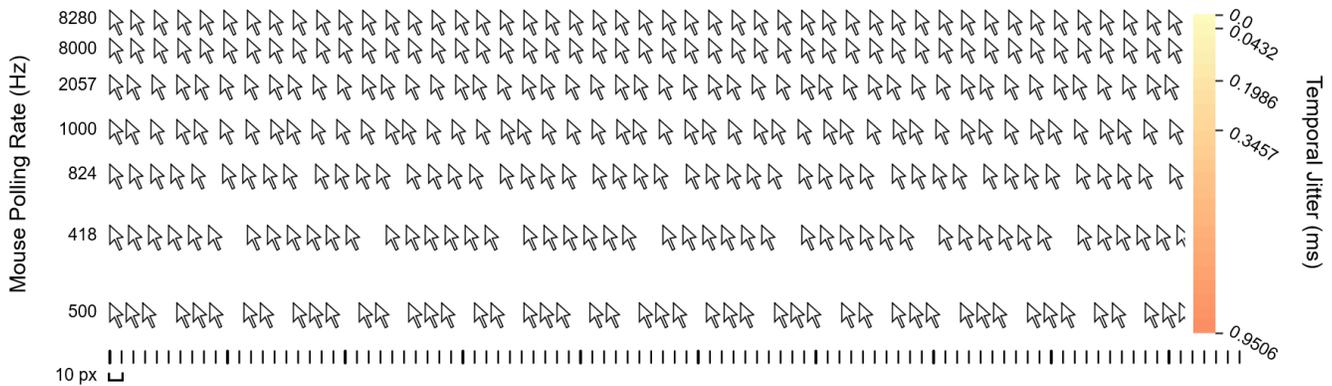


Figure 1: For same display frequency (360 Hz) and cursor speed (7000 px/s), input polling rate can determine size of spatiotemporal jitter in interactive systems. Each minor tick is 10 px.

ABSTRACT

In gaming, accurately rendering input signals on a display is crucial, both spatially and temporally. However, the asynchronicity between the input and output signal frequencies results in unstable responses called *jitter*. A recent research modeled this jitter mathematically [2]; however, the effect of jitter on human performance is unknown. In this study, we investigated the empirical effect of asynchronicity-induced jitter using a state-of-the-art high-performance mouse and monitor device. In the first part, perceptual user experience under different jitter levels was examined using the ISO 4120:2004 triangle test protocol, and a jitter of over 0.3 ms could be perceived by sensitive subjects. In the second part, we measured the pointing task performance for different jitter levels using the ISO 9241-9 (i.e., Fitts' law) test, and found that the pointing performance was unaffected up to a jitter of 1 ms. Finally, we recommended display and mouse combinations based on our results, which indicated the need for a higher mouse polling rate than that of the current standard 1000-Hz USB mouse.

*Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

UIST '21, October 10–14, 2021, Virtual Event, USA

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8635-7/21/10...\$15.00

<https://doi.org/10.1145/3472749.3474783>

CCS CONCEPTS

• Human-centered computing → HCI theory, concepts and models; Interaction devices; Pointing.

KEYWORDS

jitter, frequency, resampling, polling rate, display refresh rate, performance, game

ACM Reference Format:

Auejin Ham, Junsu Lim, and Sunjun Kim. 2021. Do We Need a Faster Mouse? Empirical Evaluation of Asynchronicity-Induced Jitter. In *The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21)*, October 10–14, 2021, Virtual Event, USA. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3472749.3474783>

1 INTRODUCTION

When controlling a continuous movement in an interactive system, e.g., movement of an object on a screen using a mouse device, it is best to render the input device signal to the output image with zero latency at an infinite frame rate, which is clearly impossible. USB input devices and HDMI (or similar) display devices process signals at their frequencies, and signals have to be paused at each stage of the transmission pipeline. Most USB mouse devices report input signals at 125, 250, 500, and 1000 Hz, which is not problematic for a common 60-Hz display because the input signal polling produces a relatively shorter delay than the display interframe delay. However, as the input and output frequencies become similar with faster monitors—modern high-frequency displays refresh their screen at 240 Hz and 360 Hz—the asynchronicity between their signal frequencies introduces unavoidable temporal and spatial jitters

[2] in the mouse cursor. As shown in Figure 1, an inappropriate combination of input–output frequencies may result in a skipping cursor trajectory. For example, a combination of a 125-Hz mouse and a 240-Hz display refresh rate will introduce a temporal jitter of approximately 4 ms. Although 4 ms does seem very long, professional First-Person Shooter (FPS) players achieve extremely fast mouse movements, such as 6950 px/s and 11583 px/s [21], which are converted into spatial jitters of 27.5 px and 46.3 px, respectively, and result in a stumbling cursor movement.

A high polling rate of a USB signal line is already known to improve the temporal performance of devices [27]; however, a polling rate of only up to 1000 Hz has been tested. Recently, a USB mouse with an 8000 Hz polling rate was introduced and has been examined in gaming communities. The main advantage of such a fast polling is a reduction in the jitter. For example, a 1000-Hz mouse on a 240-Hz display, which is common in a gaming computer setup, has a jitter of approximately 0.3 ms (= 1.4 px with a 5000 px/s cursor speed). The abovementioned state-of-the-art mouse (8000 Hz) and display (360 Hz) combination reduces the temporal jitter to under 0.04 ms and the spatial jitter to be in the subpixel range. The improvement on the faster devices is prominent in numbers, but how much are they on the human side? It is not known how high mouse and display speeds need to be. Moreover, in-depth analysis of the perceptual threshold of jitter has been limited.

This study contributes to improving the understanding of the effect of different input and output frequencies on spatial jitter and investigates the perceptual jitter threshold. First, we introduced a spatial jitter model [2] to predict the jitters for a range of mouse polling rate and display frequency combinations, and validated the model in our context (i.e., a desktop PC environment with a high-frequency polling rate input device). Subsequently, we implemented a testbed with the fastest mouse–display setup available on the market and validated its accuracy against the proposed model. Following this, we conducted a modified version of the ISO 4120:2004 triangle test with 23 participants for analyzing the perceptual threshold of jitter. We also conducted the ISO 9241-9 Fitts' law test with another 13 participants for analyzing the effect of jitter on the pointing performance. Our study showed that higher polling rates beyond 1000 Hz are beneficial for jitter reduction; however, its perceptual threshold was measured to be approximately 2000 Hz.

2 RELATED WORK

2.1 Jitter

Jitter is defined as an irregularity in a periodic signal. For example, when moving a mouse at a constant speed, we expect a perfect equidistant sequence of cursor positions. However, in reality, the distances may be irregular, as shown in Figure 1. Jitter in interactive systems is mostly caused by the sensing error of the raw input, scaling factors and nonlinear filters in the data pipeline, and tremor of human limbs [7, 22]. Even in the absence of the above factors, the asynchronicity between the input and output signal frequencies can induce a spatiotemporal jitter [2], which we focus in this study.

A large jitter degrades the task performance. For example, Batmaz et al. [3] studied the effect of rotational jitter on a three-dimensional body tracking system using different target selection methods. They found out that a jitter beyond the range of $\pm 1^\circ$ can

significantly degrade the performance (lower the speed and accuracy). Pavlovych et al. [22] conducted an in-depth investigation of the effects of spatial jitter (0, 4, 8, 16, 20 px) of an imprecise tracking system and latency (33, 58, 83, 108, 133 ms) on mouse pointing speed and accuracy based on ISO 9241-9. They neglected the effect of asynchronicity-induced jitter because its scale was extremely small to affect their results. Instead, they applied an artificial jitter to the cursor using a uniformly distributed noise. They found that the jitter had significant effects on both the throughput and error rate. They also showed that the impact of jitter on the error rate drastically increased as the target became smaller. However, it is difficult to apply the above results to asynchronicity-induced jitter because their apparatus (polling rate of 125 Hz, display frequency of 60 Hz) is outdated from the current scenario and the scale of the spatial jitter was extremely large from our perspective.

Jitter not only affects the pointing task performance [22] but also the subjective preference and the perception in a touch interaction [19]. Extrapolation of the input signal, as one of the standard methods to deal with the latency in a system, can induce spatial jitter. It originates from the extrapolation errors from the current trajectory [18], noise of the input stream [19], and quantization effects of the spatial input from the mouse or finger [19]. The amount of perceptual impact by jitter can vary in different interaction contexts. For example, the perceptual effect of spatial jitter while dragging and panning is known to be stronger than during drawing [19].

2.2 Impact of Temporal Latency and Display Sampling Rates on Performance

When using interactive devices, the end-to-end temporal latency [9, 12, 14, 17], sampling rate, and refresh rates of the input–output devices [3, 4, 8, 19, 22, 24] affect the task performance.

A high end-to-end latency in a desktop environment increases the index of difficulty of a given task, and a latency over 75 ms drastically decreases the user performance [17]. Latency can also impact the performance in moving target selection [9]. Although latency has a much stronger effect on the performance of two-dimensional (2D) and three-dimensional (3D) pointing tasks than a low-level jitter, an abrupt increase in the jitter (i.e., spikes) significantly decreases the task performance [24]. Friston et al. [12] studied the effects of latency on steering and pointing tasks. They found out that latency of 16 ms or higher can affect the performance of both tasks and that this effect is nonlinear.

A low display sampling rate also decrease the task performance. When a display frame rate is less than 15 Hz, the impact of the frame rate on the game performance is significantly higher than the impact of the display resolution [8]. Frame rate affects the performance for moving target selection more than latency [15].

Therefore, low-latency and high sampling/refresh rate devices are considered to contribute to achieving high task performance. Increasing the USB polling rate of a pointing device can effectively reduce the temporal latency of the input [27]. Baudisch et al. [4] focused on the problem of cursor jumping from one position to the next, and improved the performance of a Fitts' law task by up to 7% by extrapolating long cursor displacements. Ivkovic et al. [14] showed that the local latency (ranging from 23 to 243 ms) in 3D shooter games can have a negative effect on the performance. They

also proposed a lag compensation method that could reduce the negative impact of lag on the tracking time, acquisition time, and acquisition errors.

2.3 Human Perception Limitation

When is an interactive system considered to be sufficiently fast? The limit of human perception should be the end point. Several studies have suggested that users have different perceptual thresholds of latency in different interaction contexts. For normal touch tapping operations, Jota et al. found out that latency above 24 ms was detectable, whereas latency below 20 ms could not be perceived by any participant [16]. For touch dragging actions with a following target, the least amount of perceivable latency is reported as 5–10 ms [20]. In addition, in touch dragging, a change in latency is perceivable when it exceeds 8.3 ms [10]. While performing comparatively more complex tasks such as writing and drawing, temporal latency smaller than 50 ms is not perceivable [1]. The just noticeable difference (JND) of the perceivable latency was found to be approximately 96 ms for tapping and 55 ms for dragging tasks using indirect touch devices.

3 TEST BED STUDY

Unlike the theoretical Jitter model of Antoine et al. [2] that only considers input device polling rate and the display refresh rate, a real-world system consists of additional components, e.g., an operating system messaging pipeline, signal processing latencies, and competing applications for resources. We assumed that the mouse polling rate and the display refresh rate are the two dominating factors of a jitter in our context; however, those additional components may further worsen the jitter. Therefore, we recorded the real cursor trajectories with high-resolution timestamps and verified that our implemented testbed correctly followed the model of Antoine et al. by comparing the theoretical and measured jitter sizes. We concluded that our assumption was plausible and that our testbed yielded the expected amount of spatial jitter reasonably.

3.1 Jitter Model

Antoine et al. [2] proposed a theoretical model to calculate the expected sizes of the temporal and spatial jitters induced by asynchronous input and output signal rates. Their model calculates the expected mean of the temporal jitters using a constant polling rate (F_i) and a constant display frequency (F_d). However, because their research focused on a mobile touch interface whose input stream was extrapolated, they empirically measured the spatial jitters, instead of predictively modeling the instantaneous input speed. Because we focus on a desktop environment whose input signal is not resampled, we assume the expected mean of the spatial jitters is the expected mean of the temporal jitters multiplied by the cursor speed. Regarding our testbed study, first we briefly explain the model of Antoine et al. and present the validation of our assumption of a uniform cursor speed by comparing the results of the model with the spatial jitters from simulated trajectories.

Let us denote the polling rate of an input device as F_i and the display frequency to render as F_d . We denote the i -th input event time as $t_i[i]$ and the j -th display update time as $t_d[j]$. If $F_i \neq F_d$,

$t_i[x] \neq t_d[x]$ for any $x > 0$. Therefore, we denote $k(j)$ as the last input event before the j -th display render. Temporal jitter is defined as the mean of the absolute differences of the asynchronicity-induced lag, $L[j]$, between time $t = t_d[j]$ of the j -th output event and time $t = t_i[k(j)]$ of the latest input event prior to $t_d[j]$.

Assuming that both the display frequency (F_d) and mouse polling rate (F_i) are constant, the temporal jitter can be generalized and obtained only from the two frequencies: $\overline{\Delta L} = \frac{1}{N} \sum_{j=0}^N |\Delta L[j]| = \frac{2a(1-a)}{F_i}$ with $a = \frac{F_i}{F_d} - \lfloor \frac{F_i}{F_d} \rfloor$. To calculate the spatial jitter from the model of Antoine et al., first we calculate a difference vector $D(t)$, which is the displacement between the positions in the input signal, $P_i(t)$, and in the resulting output signal, $P_d(t)$. The difference vector represents the cursor displacement error for each updated frame on the display. Therefore, similar to the temporal jitter, we can obtain the expected value of the asynchronicity-induced spatial jitter over a trajectory as the mean of the absolute differences: $\overline{\Delta D} = \frac{1}{N} \sum_{j=0}^N |\Delta D[j]|$. To calculate the difference vector, the model estimates the true input signal, $P_i(t[j])$, by linearly interpolating the previous raw input, $P_i(t[j-1])$.

We follow the assumptions defined by Antoine et al. [2]: (1) the display frequency (F_d) and the polling rate (F_i) are constant, (2) no spatial transformation (i.e., C-D gain) is applied, and (3) “sensing, computation, rendering, and display are instantaneous.” Therefore, in our scenario, each mouse event polled between two display refreshes is rendered. Considering the nature of Windows HID (Human Input Devices) architecture on which our experiments are conducted, we also assume the display to show the most recent cursor displacement with no application of resampling, i.e., $P_d(t = t_d[j]) := P_i(t = t_i[k(j)])$.

3.2 Cursor Speed and Temporal and Spatial Jitters

We measured spatial jitters ($\Delta D[j]$) from a cursor trajectory simulation and used them to validate the assumption ($\overline{\Delta L}$) from the model of Antoine et al. First, we generated 10 s of uniform cursor motion trajectories by simulating constant mouse polling rates and display frequencies as explained in Section 3.1. We simulated different mouse polling rates in the range of 125–8000 Hz and display frequencies in the range of 60–360 Hz. We considered a controlled cursor speed of 100 px/s or 100 counts/s as we assumed a 1:1 C-D gain from the mouse sensor to the screen pixels. For simplicity, we assumed there is no starting latency difference between the cursor and display logs. We measured the mean size of spatial jitters divided by the constant cursor speed for each simulated trajectory (see Figure 2a). Subsequently, we computed the expected size of temporal jitter using the model of Antoine et al. (see in Figure 2b). Both figures visually illustrate a uniform distribution of the jitters.

Using the same simulated trajectories, we also compared the measured temporal jitters ($\Delta L[j]$) with the assumption ($\overline{\Delta L}$) of the model for each frequency pair (jitter conditions) later to be used in our user studies (see Table 1). The mean absolute percentage error of the model were 0.06755% (SD = 0.07594) for 240-Hz display frequency and 0.02507% (SD = 0.02873) for 360 Hz.

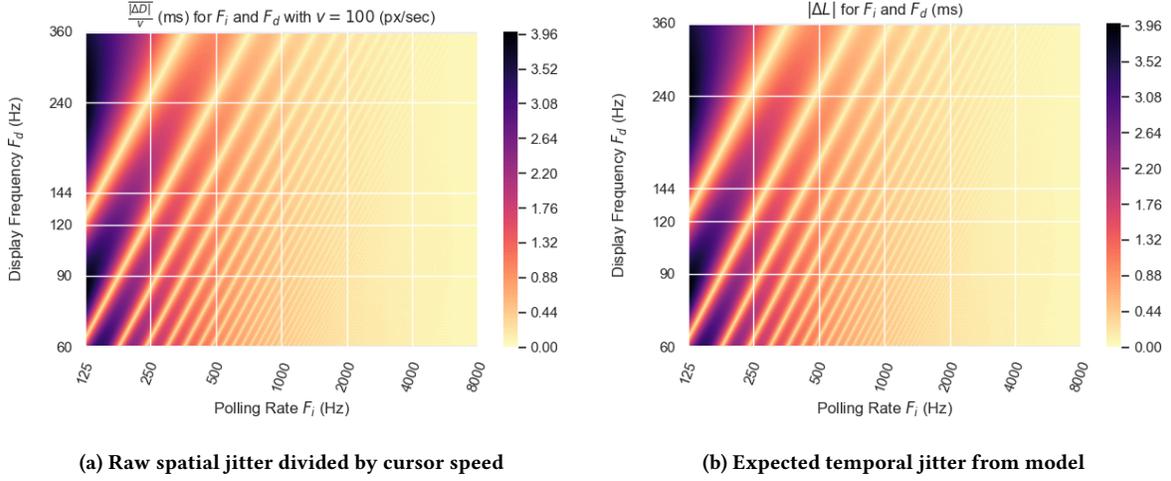


Figure 2: Under uniform mouse speed, spatial jitter divided by cursor velocity yields temporal jitter.

Table 1: Raw temporal jitters (ms) from uniform cursor trajectories ($v = 100$ px/s, duration = 10 s).

F_d (Hz)	240					360				
F_i (Hz)	8000	4000	2000	1000	500	8000	4000	2000	1000	500
model	0.05555	0.11111	0.22222	0.27778	0.30556	0.04321	0.04938	0.24691	0.34568	0.95062
mean	0.05554	0.11112	0.22218	0.27755	0.30492	0.04320	0.04934	0.24692	0.34571	0.95054
SD	0.01964	0.03929	0.07855	0.24825	0.45968	0.02886	0.06105	0.02761	0.23099	0.21665

3.3 Spinning-Poller Logger

Obtaining high-resolution timestamps for polling and displaying events is essential for calculating the spatial jitter in real cursor trajectories. We developed a logger software on Processing¹ utilizing spinning-poller (i.e., busy waiting) method to collect mouse events. Using our system, mouse data and displayed cursor location are recorded by a thread spinning in a controlled display frequency. Unlike the model of Antoine et al. [2] in which the polling rate is assumed to be constant, our spinning poller collects cursor data for each spin loop on a designated thread, enabling cursor data to be polled as soon as they are updated.

We set the initial temporal delay from the first input event ($t_i[0]$) to the first display update ($t_d[0]$) as zero. The latest mouse input signal to be rendered in the j -th output of the display is generated at time $t_i[k(j)]$. We denote the resulting output of the cursor location as $P_i(t = t_i[k(j)])$. For each spin, the thread updates its internal clock as the temporal duration from its initialization in nanoseconds. When the cursor location is changed from the previous spin, the current timestamp, t_i , and the most recent cursor location, $P_i(t = t_i)$, are overwritten on $t_i[k(j)]$ and $P_i(t = t_i[k(j)])$, respectively, in the stack after recording the polling time ($dt_i[k(j)] := t_i[k(j)] - t_i$). For each display update at a fixed frequency, the thread writes its internal clock into $t_d[j]$. Subsequently, each row consisting of $t_d[j]$, $t_i[k(j)]$, $dt_i[k(j)]$, and $P_i(t = t_i[k(j)])$ is recorded. While conducting our first user study, the experiment status (index of

the triad, index of the label orders for each triad, and polling rate condition) is also recorded for each row.

3.4 Accuracy Validation

The implemented logger is working on Java Runtime Environment (JRE), which is considered slower than a native binary application. To validate the implemented logger is achieving the desired performance, we compared our logger against MouseTester², a low-level event hook-based poller for recording Windows HID messages directly. For each HID stack update, MouseTester records the x and y displacements of a mouse in units of counts with a timestamp (ms) from the initiation. We executed both our spinning poller and MouseTester and recorded real cursor trajectories simultaneously. Subsequently, we measured the instantaneous polling rates and the temporal jitters using all trajectories. We used the same apparatus as that in our user studies (see Section 4.1.1). Under each polling rate condition, 66.12 s of cursor trajectories (SD = 5.542) were recorded. The average number of samples was 113903.4 (SD = 61308.84) for MouseTester and 18880.0 (SD=5065.95) for our spinning poller. We anticipated a large difference between the sample sizes of the two algorithms, even with simultaneous logging, because our spinning poller records for each display refresh, whereas MouseTester records for each HID stack update. When a mouse reads a displacement value smaller than 1 count per inch (CPI), the value is not reported but accumulated until the sum is over 1

¹<https://processing.org>

²<https://github.com/microe1/MouseTester>

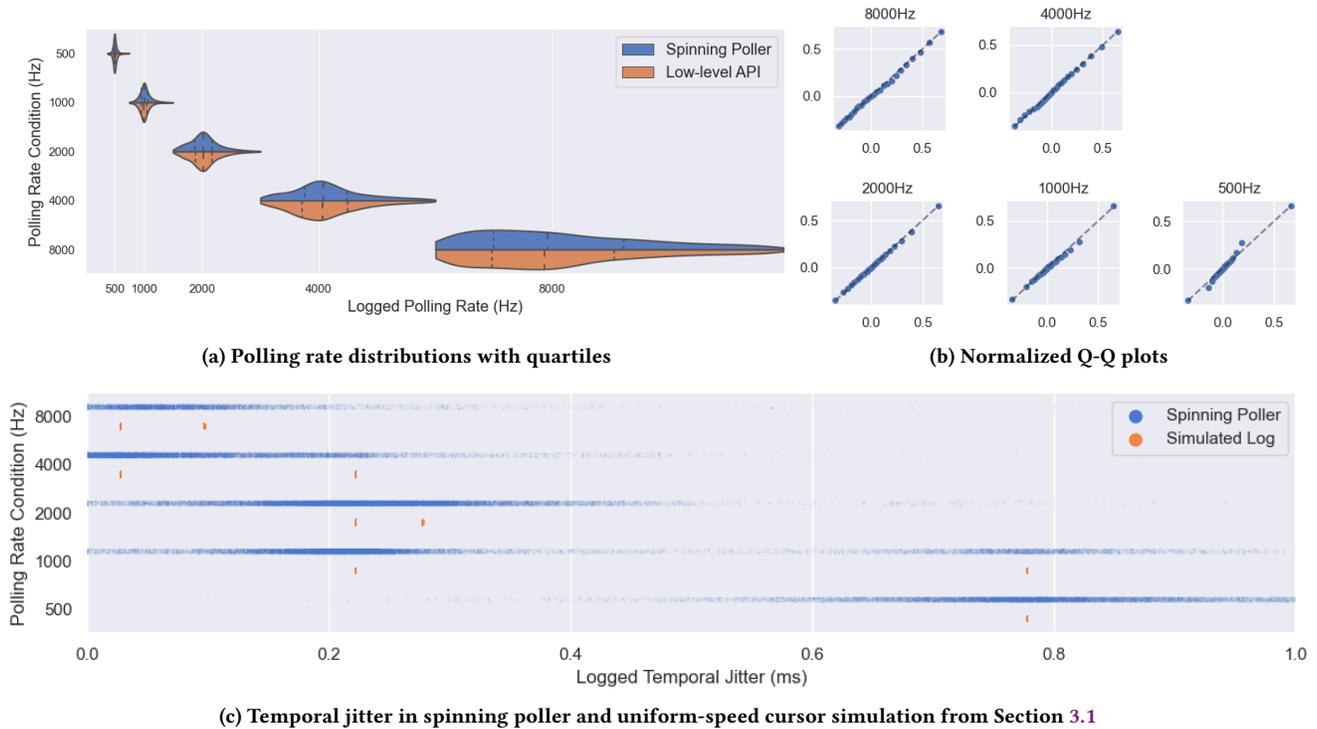


Figure 3: Polling rate and temporal jitter of spinning poller are similar to those of low-level API logger and cursor simulator.

count. Therefore, an F_i -Hz mouse can report its signal in N/F_i s of time ($1 \leq N$). Because we aimed to compare the performance of the two logging systems under the highest stress (mouse speed), samples with temporal intervals longer than $1.5/F_i$ were excluded. We validated the temporal accuracy of our spinning poller using the measured polling rates and temporal jitters. The polling rate distributions from both logging systems are illustrated in Figure 3a. Each normalized Q-Q plot with 27 percentile points in Figure 3b shows the similarity in the polling rate distributions obtained from the two recording algorithms.

To validate the accuracy of the measured temporal jitters, we simulated uniform-speed cursor trajectories, as described in Section 3.1. However, the recording times and cursor speeds were assigned as the mean value of the corresponding trajectory obtained using our spinning logger. We compared the temporal jitters measured by our system and using the simulated trajectories. Figure 3c shows that the distributions of the recorded temporal jitters using our spinning poller are centered around the temporal jitters of the simulated trajectories.

The results in Figure 3 show that the overall fidelity of the Processing logic (with Java backend) is sufficiently high, being comparable to RAWHID API, which is considered the most accurate information available on the OS level. Thus, we concluded that the overall apparatus on Java was sufficiently fast and rendered the overall cursor jitter characteristics correctly.

4 USER STUDY: TRIANGLE TEST

If one can perceive the difference in jitters, a task that could benefit from it could be identified. Therefore, we measured the JND to assess the perceivable smoothness of the cursor qualitatively. In addition to the implemented testbed system, we designed a user study based on a modified version of the ISO 4120:2004 triangle test. The objective of this user study was to obtain the perceptual discrimination of different spatial jitter levels introduced by the asynchronicity between mouse and display frequencies.

4.1 Experimental Design

Because the mouse had limited polling rate options, we could not adopt the JND experiment procedure, which requires a gradual change in the jitter sizes. To compensate the limited choice of stimuli granularity, we performed the ISO 4120:2004 triangle test [13], one of the most frequently used discrimination tests for determining whether participants can distinguish two different stimuli. A triad is composed of three stimuli: two are identical and one is different. A participant is forced to choose one different stimulus. We chose two available high display frequencies (240 Hz and 360 Hz) at which the temporal jitter becomes extremely prominent (Figure 2). Subsequently we set the reference condition as the best condition at each display frequency, i.e., a mouse polling rate of 8000 Hz. Under a given display frequency, the participants were asked to distinguish the reference 8000 Hz polling rate from four lower polling rates (500, 1000, 2000, and 4000 Hz) using the triangle test procedure. The participants examined a white cursor moving on a black full-screen display, focusing irregularly on the cursor

trajectories. Our experiment was designed in a mixed manner with the display frequencies as between-subject factors and the polling rates as within-subject factors. This was because different display frequencies introduce more noticeable differences in the cursor movement than the effect of jitter. For each display frequency, a triangle test was conducted with a total of 120 triads (4 polling rates \times 6 possible order combinations³ \times 5 blocks). The presentation order of the four polling rates was randomized, and the triads within a test were also presented in a random order.

4.1.1 Apparatus. The experimental environment consisted of a desktop, mouse (Razer Viper 8k, 800 CPI, maximum 8000 Hz polling rate), mouse pad (350 mm X 320 mm), 24.5-inch display (ASUS ROG SWIFT PG259QN, 543.17 mm X 302.62 mm, 1920 px X 1080 px, maximum 360 Hz refresh rate), and keyboard. The display could work under different native refresh rates, and G-SYNC was enabled. Using a high-speed camera (960 FPS), we measured the button-to-pixel latency of our apparatus, which was 17.97 ms (SD = 1.860 ms). The C-D gain was set as 1:1, where 1 count from a mouse signal was translated into 1-px movement on the display. A separate laptop was used for recording the triangle test answers. The mouse pad was located on the left of the keyboard for a short hovering range of hands.

We developed a graphic interface for visualizing the cursor and the triad labels using Processing. Over a full-screen black background, a software showed a white square cursor (3 mm X 3 mm), the condition counts, the triad counts, and three labels to choose from each triad (see Figure 4). The labels of the selected stimuli were shown as twice larger than the other two labels. When the participants switched the stimuli, the software would hide the square cursor, stop the cursor logging process, and run an automated macro on a dedicated mouse driver software on the second monitor hidden from the participant. Within the second monitor, the macro clicks the desired polling rate settings and relocates the cursor to its previous position. The participants were forced to use only one hand during the experiment. This rule prevented unexpected cursor movement while the macro was running as they could not press keys without leaving the mouse. After the macro execution was finished, a separate notification appeared that requested the participants to shake the mouse to invoke the cursor. During this procedure, a quick sanity check was performed to ensure the desired polling rate was set.

4.1.2 Participants. Twenty three right-handed participants (19 males and 4 females, age from 19 to 31, average age 25.5) were recruited from a local university. Twelve (2 female, 10 male, average age 25.16) tested a 360-Hz display, and eleven (2 female, and 9 male, avg. age 25.91) tested a 240-Hz display. The experiment lasted approximately 40 min, and each participant was reimbursed 10,000 KRW (\approx 9 USD) for the participation. The Internal Review Board (IRB) of the university approved the study procedure and data collection.

4.1.3 Procedure. The participants signed an informed consent forms and filled an online demographic survey. They sat in a calm and dimmed office room. An experimenter explained the task to

³When constructing triads with two stimuli A and B, the six possible orders were (ABB, BAB, BBA, AAB, ABA, and BAA)

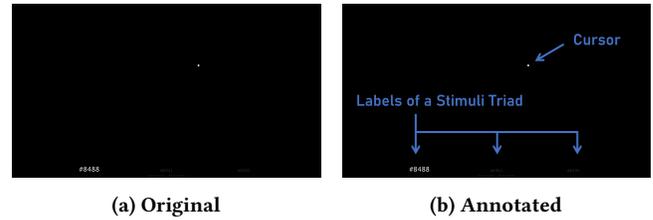


Figure 4: Interface of triangle test observed by participants.

the participants. For each mouse polling rate, 30 triads were tested. The three stimuli in each triad were presented as a set, labeled randomly as four-digit numbers. Participants pressed A or D key on the keyboard to switch to the next stimuli. The participants could examine the stimuli to the maximum extent, and they were forced to choose one stimulus that was different from the two others. The correct answer was not provided throughout the experiment.

4.2 Results

We collected total 2,760 selections from each stimulus triad, with 1,440 selections made under 360-Hz display and 1,320 selections under 240-Hz display. We also collected mouse events while running the experiment. We excluded zero-distance events and events with a temporal interval longer than $1.5/F_i$, which originated from a slow cursor movement. Among the 20,199,880 events collected throughout the experiment, 5,872,891 samples were analyzed.

4.2.1 Demographic Survey. We asked participants' everyday computer peripheral device settings, their preferred games, and their rankings in games. As mouse polling rates, 13 participants used 1000 Hz, 4 used 125 Hz, three used 500 Hz, and 1 used 90 Hz. The mouse polling rates of two participants were unknown. As the display refresh rates, 10 participants used 144 Hz, 9 used 60 Hz, and 70, 120, 165, and 240 Hz were used by 1 participant. The participants used their monitors for 6.217 h daily (SD = 3.671). Among the 21 participants who reported playing games on a PC, 14 participants stated that they play FPS games such as PUBG, Overwatch, or Apex Legends Gold for 7.857 h weekly (SD = 7.150). Among them, 8 participants were ranked in the top 20% players on each server. Thirteen participants reported that they play League of Legends, a multiplayer online battle arena game, for 5.385 h weekly (SD=5.212). Among them, 3 participants were ranked in the top 20% players. Overall, 9 participants were ranked in the top 20% players on one or more games. We hereafter call them "rankers." Approximately 77.8% of the rankers (N = 7) used a 1000-Hz mouse and 77.8% of the rankers (N = 7) used displays with a refresh rate of 144 Hz or higher for their desktop environments. Using the model of Antoine et al. [2], we obtained the expected sizes of the temporal jitters in the environment of each participant (mouse polling rate and display frequency). The average temporal jitter in the environment of each participant was 0.708 ms (SD = 1.163). The environments of the rankers caused a temporal jitter of 0.193 ms (SD = 0.178), whereas of the non-rankers of 1.024 ms (SD = 1.379).

4.2.2 Distinguishable Size of Spatial Jitter. Owing to the nature of the triangle test, the number of correct answers for each stimulus pair follows a chi-square distribution. Therefore, within 30 triads,

Table 2: Upper section shows expected values from model of Antoine et al. along with medians of measured temporal and unit-speed spatial jitters. Lower section shows triangle test scores by jitter conditions. Note that counts are cumulative in different α levels (e.g., $\alpha < 0.05$ row includes results of $\alpha < 0.01$ and $\alpha < 0.001$).

F_d (Hz)	240					360				
F_i (Hz)	500	1000	2000	4000	8000	500	1000	2000	4000	8000
model (ms)	0.306	0.278	0.222	0.111	0.056	0.951	0.346	0.247	0.0494	0.043
raw $ \Delta L $ (ms)	0.186	0.195	0.199	0.084	0.065	0.854	0.227	0.223	0.058	0.064
raw $ \Delta D /v$ (px)	0.228	0.214	0.224	0.093	0.077	0.894	0.243	0.230	0.078	0.078
$\alpha < 0.001$	4/11	1/11	0/11	0/11		7/12	0/12	0/12	0/12	
$\alpha < 0.01$	4/11	1/11	0/11	0/11		8/12	0/12	1/12	0/12	
$\alpha < 0.05$	4/11	1/11	0/11	0/11		8/12	0/12	2/12	0/12	
avg. score	15.27	10.82	11.00	9.09		21.33	9.25	11.33	9.25	

the number of the correct answers above 15 indicates the participant can distinguish two stimuli with false positive rate $\alpha < 0.05$ ($3.841 < \chi^2(16)$), 16 answers with $\alpha < 0.01$ ($6.635 < \chi^2(17)$), and 18 answers with $\alpha < 0.001$ ($10.828 < \chi^2(19)$). The average score of the triangle test under each condition is provided in Table 2. Over half of the participants ($N = 12$) could distinguish 500 Hz and 8000 Hz ($\alpha < 0.001$ for $N = 11$, $\alpha < 0.01$ for $N = 12$). One participant could separate 1000 Hz from 8000 Hz with $\alpha < 0.001$. Two participants could distinguish 2000 Hz and 8000 Hz with $\alpha < 0.01$ and $\alpha < 0.05$, respectively. However, these results are estimated to have occurred coincidentally because the two participants could not perform the easier task of distinguishing 1000 Hz and 8000 Hz. These results showed that for a 240 or 360-Hz display, the perceivable difference in the jitters of the 1000-Hz mouse and 8000-Hz mouse was not significant for the majority of participants, whereas the difference was significant for the 500-Hz mouse.

We assume if people can distinguish a given size of jitter, they can also distinguish a larger one. From the jitters using the 8000-Hz mouse (0.0432 ms with $F_d = 360$ Hz, 0.0556 ms with $F_d = 240$ Hz), over half of the participants ($N = 12$) could distinguish a 0.951 ms or smaller jitter ($\alpha < 0.001$ for $N = 11$, $\alpha < 0.01$ for $N = 12$). Specifically, four participants could distinguish a 0.306 ms or smaller jitter ($\alpha < 0.001$). One participant could distinguish a jitter of 0.278 ms ($\alpha < 0.001$). Over half of the participants ($N = 12$) could distinguish jitters smaller than that in the average non-ranker environment, whereas no participants could distinguish the jitters in the average ranker environment. This implies that the environments of the current rankers might be sufficient for minimizing perceptible jitters.

We also compared the triangle test scores of the 9 rankers and the 14 non-rankers. Among them, 5 rankers and 7 non-rankers used a 360-Hz display, and 4 rankers and 7 non-rankers used a 240-Hz display. Approximately 22.2% of the rankers ($N = 2$) could distinguish at least one temporal jitter (0.951, 0.306 ms) from the jitters of the 8000-Hz mouse (0.043, 0.056 ms). Approximately 71.4% of the non-rankers ($N = 10$) could distinguish at least one temporal jitter ($N = 7$ with 0.951 ms, $N = 2$ with 0.306 ms, and $N = 1$ with 0.278 ms). Wilcoxon’s signed-rank test showed that no significant difference in the triangle test scores of the rankers (avg.=35.5%) and non-rankers (avg.= 43.1%), $W = 7.000$, $p = 0.529$. This shows that



Figure 5: Expected jitter sizes (unit: ms) for possible combinations of mouse polling rates and display frequencies. Red cells are bad and green cells are good jitter sizes, based on perceptual jitter threshold found in our experiment.

highly skilled game players are not necessarily more sensitive to the size of jitter.

4.2.3 Cursor Speed and Temporal and Spatial Jitters. Under each jitter condition, Figure 6 shows the median of the raw temporal jitters (yellow circle) as the median of the raw spatial jitters divided by the cursor speed (green triangle) and the expected sizes of the temporal jitters by the model of Antoine et al. [2] for the given frequency pairs (red cross). From the median of the raw temporal jitters (yellow circle), the mean percentage error of the model of Antoine et al. [2] is 28.75% (SD = 18.38) and that of the linear model is 14.53% (SD = 9.254) under each condition. The median slope of each heat map (dotted red line) is 1.057 on average (SD = 0.04451), showing that the raw temporal jitters of the real cursor trajectories are similar to the sizes of the raw spatial jitters divided by the cursor speed. Similar to the testbed study described in Section 3.2, the relation between the spatial and temporal jitters holds even with the real cursor trajectories. Therefore, we can measure the perceivable threshold of the temporal jitter to obtain that of the spatial jitter, which is visually represented as irregular cursor intervals over a unit-speed trajectory.

4.2.4 Recommended Display and Mouse Combinations. From the results, we assumed that a temporal jitter of over 0.3 ms could be perceived by some sensitive users, whereas a jitter of less than 0.2 ms would be imperceptible for most people. Applying these

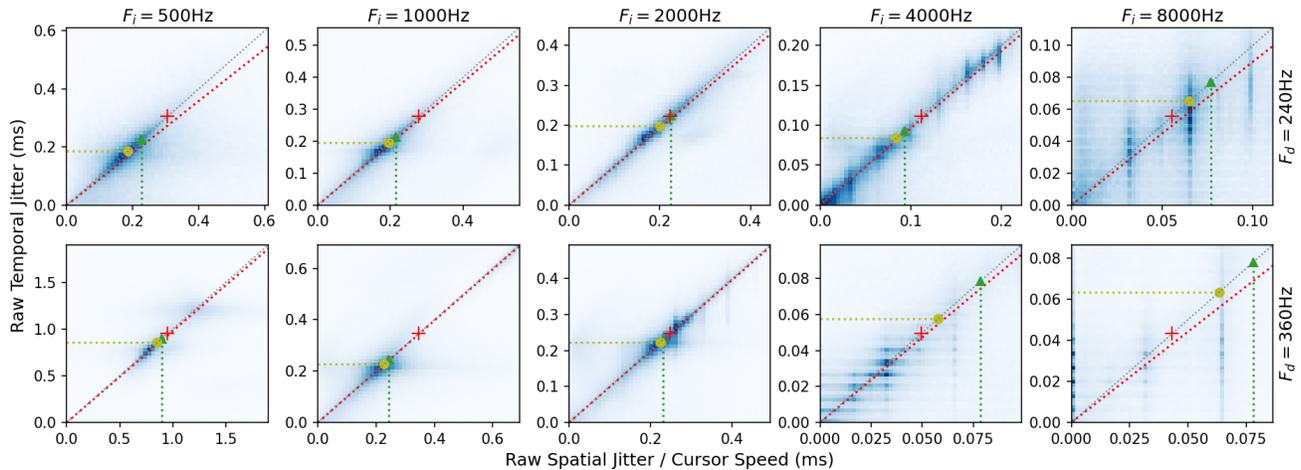


Figure 6: Density heat maps of unit-speed spatial and temporal jitters (each median as green triangle and yellow circle), median slope between them (dotted red line), and expected sizes of temporal jitters from model of Antoine et al. (red cross).

thresholds to the jitter model introduced in Section 3.1, we tabulated a set of *good* and *bad* display and mouse combinations. As shown in Figure 5, mouse polling rates less than or equal to 500 Hz introduce noticeable jitters at all display frequencies, which are supported by our experiment results. Mouse polling rates more than 2000 Hz are clearly good. A 1000 Hz mouse polling rate, the de-facto standard limit of a USB mouse, is interesting because the jitter fluctuates on variation of the display frequencies. Even though our experiment results exhibit 1000 Hz mouse is mostly indistinguishable from 8000 Hz mouse, we consider 1000 Hz as the borderline of the human perceptual jitter recognition.

5 USER STUDY: FITTS' LAW TEST

To examine the effect of the jitter level on the pointing performance quantitatively, we conducted the ISO 9241-9 standard test (2D Fitts' Law test). New ranker participants were recruited, who are (or were) ranked in the top 20% players of games demanding extensive mouse control skills.

5.1 Experiment Design

The experiment used a 2×5 within-subject design with two factors: display frequency (240 and 360 Hz) and mouse polling rate (500, 1000, 2000, 4000, and 8000 Hz). Under each jitter condition, 18 combinations of three amplitudes (256, 384, and 512 px) and 6 target widths (8, 16, 32, 64, 96, and 128 px) were set, and resulting indices of difficulty (*ID*) range was 1.585–6.0224. Consequently, 2 (display) $\times 5$ (mouse) $\times 18$ (A^*W) = 180 datapoints from 3,600 mouse clicks were collected per subject. Separately, the same logger as mentioned in Section 3.3 collected raw mouse movements in background. The presenting order of the jitter conditions was randomized.

5.1.1 Apparatus and Procedure. We conducted the same pre-experiment survey in our triangle test to collect demographic data. We also used the same jitter conditions and apparatus (a mouse,

mouse pad, 24.5-inch display, and keyboard); however, we set both polling rates and display frequencies as within-subject factors of this experiment. We used Wobbrock et al.'s FittsStudy⁴ to collect the data and calculate the throughput (more specifically, TP_{avg} [28]). The participants sat in a calm and dimmed office room and wore form earplugs during all phases of the experiment. Before each experiment, they had unrecorded sessions of FittsStudy trials until they felt adapted to the given mouse shape and CPI.

5.1.2 Participants. A total of 13 participants (13 males, age from 18 to 25, avg 21.63) from a local university signed their consent forms and agreed to participate in this study. The experiment lasted approximately 40 min. They were reimbursed 15,000 KRW (≈ 13.5 USD) for their participation. Among the 10 participants who played League of Legends, 3 were ranked in the top 5–10%, 3 in the top 1–5%, and 3 in the top 1%. In the 2 subjects who play Overwatch, 1 was ranked in the top 5–10% and the other one in the top 5%. One subject who plays Osu!, a target-clicking rhythm game, was ranked in the top 1–5% of Korean players. Approximately 84.6% of the participants ($N = 11$) used a 1000-Hz mouse in their PC environment. Approximately 53.8% of the participants ($N = 7$) used displays with a refresh rate of 144 Hz or higher. The mean temporal jitter of their PC environments obtained with the model of Antoine et al. [2] was 0.395 ms (SD = 0.381).

5.2 Results

Figure 7a illustrates the mean of the bivariate throughputs (TP_{avg} [28]) of each participant under each condition. A two-factor repeated measures ANOVA was conducted. All factors and their interactions did not have any significant main effect on the pointing performance in the throughput: polling rate ($F_{4,48} = 0.201, p = 0.937$), display frequency ($F_{1,12} = 0.598, p = 0.454$), and their interaction ($F_{4,48} = 0.417, p = 0.796$).

⁴<https://depts.washington.edu/accelab/proj/fittsstudy/index.html>

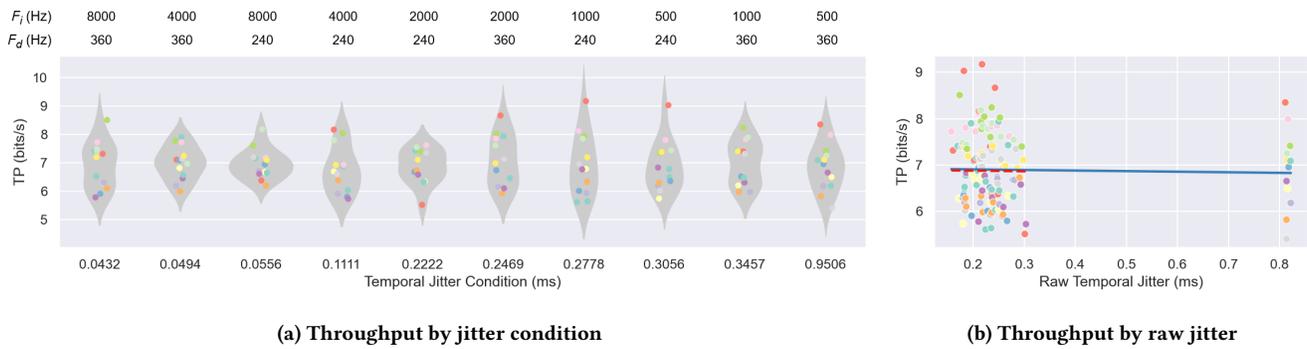


Figure 7: Throughput by temporal jitter. Each dot represents mean throughput of 360 data points under each display–mouse rate combination. Blue solid line is Spearman correlation of all data points, whereas red dashed line is Spearman correlation without worst jitter condition of 0.9506 ms.

We also conducted linear regression analysis of the jitter size and the throughput. The blue solid line in Figure 7b illustrates the Spearman correlation between the throughput and the median raw temporal jitter ($r_s = -0.124$, $p = 0.159$), whereas the red dashed line shows their Spearman correlation ($r_s = -0.131$, $p = 0.157$) without the worst jitter condition of 0.9506 ms. The results indicate a very weak negative correlation between the jitter and the throughput.

To conclude, although the range of the jitters tested in this experiment does not drastically change the overall pointing performance, we do not completely reject the need for a faster mouse device.

6 CONCLUSION

This study addressed the perceptual threshold and the performative impact of the spatiotemporal jitter induced by asynchronous input and output system frequencies. We implemented a high-definition logger to record the jitter for a given cursor trajectory, validated its temporal accuracy, and compared the measured and mathematically modeled jitter sizes. Using a state-of-the-art display and experimental high-polling rate mouse devices, we provided an extreme range of jitter levels aiming at high-end gaming environments.

First, we validated the model of Antoine et al. [2] in a real environment for the two high-end display frequencies available (240 Hz and 360 Hz) and a range of mouse polling rates (500 Hz – 8000 Hz). Our implemented system correctly produced and measured the expected amounts of jitters for the set of display–mouse rate combinations. Interestingly, a higher framerate display may introduce more jitter than a lower one in particular combinations (e.g., 360 Hz display and 1000 Hz mouse vs. 240 Hz display and 1000 Hz mouse).

In the first part, we obtained the perceptual experience under different amounts of asynchronicity-induced jitters using the triangle test protocol. Subsequently we analyzed the expected amounts of jitters in the PC environments of high-ranking game players (top 20%) and those of the other participants and compared them with the perceptual threshold of jitter. The results suggested that a temporal jitter size of <0.3 ms is expected to be imperceptible, at least in our experiment setting—a small white cursor moving on a solid black screen.

In the second part, we recruited additional high-ranking game players (top 20%) and measured their pointing task performance under different amounts of jitters using Fitts’ law test. The amount of jitter had no significant effect on the pointing task performance. A survey of the monitors and mouse performance of the gamers suggested that the average PC environment of the high-ranking game players was sufficient to suppress the jitter in the perceivable range.

Our results also offer a guideline for gaming PC environments; for example, Table 2 and Figure 5 can guide the choice of hardware frequencies to minimize the cost while maintaining the jitter below the perceivable amount. Our simulation suggested that there are some “good” and “bad” combinations of display frequencies and mouse polling rates. In general, high mouse polling rates of over 2000 Hz exhibit good performance in the entire range of the display frequencies. Also, 1000 Hz mouse exhibited a borderline temporal jitter by human perceptual system. This suggests a need for slightly higher polling rate than the current de-facto standard mouse polling rate limitation, 1000 Hz.

7 DISCUSSION

In this section, we discuss the interpretation of the results, limitations, and future studies.

7.1 Human Perception

First, we discuss the underlying human mechanism behind jitter perception. As shown in Figure 2b, the temporal jitter sizes are very small, <4 ms, even in the worst case in the practical range of display and mouse settings. This small timespan difference is imperceptible because our vision is poor in the time domain. Instead, human vision is good at a precise spatial resolution. Therefore, the jitter is expected to be perceived as an irregularity of the consequent cursor afterimages left in the retina. Thus, a high cursor speed reveals jitter prominently and may affect JND perception. However, we did not artificially control the cursor speed because it breaks the external validity of the experiment. In most cases, a cursor is controlled actively by its user. In this context, we expected the participants to examine various cursor speeds by themselves to discriminate different jitter sizes in the experiment.

Second, the performance evaluation was only done on one type of task in a short-term study. The participants were exposed to new devices and had limited time to adapt themselves to the testbed system. The statistical test could not find a significant main effect in our study; however, it does not completely disprove jitter-induced performance reduction. With a longitudinal study, participants will have an opportunity to develop particular skills benefited from a low-jitter system. In addition, we should clarify that the results for the smallest JND (≤ 0.306 ms) are based on only a small number of individuals (4) who could perceive this amount of jitter. Furthermore, our participants mainly identified their gender as male; therefore, the results have limitations in the generalization to other genders.

Finally, different cursor types and sizes may improve the jitter perception. An extreme case could be an entire screen translation from a mouse movement, such as in FPS games, which maximizes visual stimuli to the photoreceptors in the retina. The possible future studies will include obtaining a personalized JND of asynchronicity-induced jitter and optimizing the polling rate in different interfaces (e.g., head-mounted displays), stimulus modalities (e.g., tactile feedback), and interaction contexts.

7.2 Hardware Specifications

Monitor ghosting (i.e., retained images owing to the short pixel response time) may change the perceptual threshold of cursor jitter. We attempted to minimize the effect of ghosting by adopting the fastest possible monitor in the market, and therefore, our result could serve as a baseline under the optimal condition. With slower monitors, the effect of jitter may even increase, justifying the need for a faster mouse.

Under gaming gear advice and pro-gamer settings, it is generally advised to “turn off any mouse accelerations” [5, 6, 11, 23, 25, 26], which was the reason behind the choice of a constant transfer function. Different transfer function settings may affect the jitter perception.

While recording cursor trajectories in our experiments, G-SYNC was activated to reduce the screen tearing by synchronizing the display and GPU frequency. Indeed, it is a variable display frequency setting. To mitigate its effect, we set the frame rate of our triangle test software as twice of the display refresh rate to ensure the resampling nature of G-SYNC did not break the assumption of a constant display frequency. However, G-SYNC introduces a constant latency owing to the lookaside buffer (approximately 1–2 ms) and may disturb the constant display frequency assumption owing to the nature of the technology. Examining the effect of a variable display frequency remains as another future study.

ACKNOWLEDGMENTS

This work was supported by the DGIST Start-up Fund Program of the Ministry of Science and ICT (2021010011). Razer Inc. kindly provided the prototype mouse device used in the experiments without monetary compensation and did not intervene in this study.

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 2014* (Montreal, Quebec, Canada) (GI '14). Canadian Information Processing Society, CAN, 167–174.
- [2] Axel Antoine, Mathieu Nancel, Ella Ge, Jingjie Zheng, Navid Zolghadr, and G ry Casiez. 2020. Modeling and Reducing Spatial Jitter Caused by Asynchronous Input and Output Rates. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) (UIST '20). Association for Computing Machinery, New York, NY, USA, 869–881. <https://doi.org/10.1145/3379337.3415833>
- [3] Anil Ufuk Batmaz, Mohammad Rajabi Seraji, Johanna Kneifel, and Wolfgang Stuerzlinger. 2020. No jitter please: Effects of rotational and positional jitter on 3D mid-air interaction. In *Proceedings of the Future Technologies Conference*. Springer, 792–808. https://doi.org/10.1007/978-3-030-63089-8_52
- [4] Patrick Baudisch, Edward Cutrell, and George G Robertson. 2003. High-density cursor: A visualization technique that helps users keep track of fast-moving mouse cursors. In *Interact*, Vol. 3. 236–243.
- [5] Fran ois Beaufort. 2020. *Disable mouse acceleration to provide a better FPS gaming experience*. Retrieved July 05, 2021 from <https://web.dev/disable-mouse-acceleration/>
- [6] BestSettings 2019. *What Is Mouse Acceleration?* Retrieved July 05, 2021 from <https://www.bestsettings.com/library/what-is-mouse-acceleration/>
- [7] G ry Casiez, Nicolas Roussel, and Daniel Vogel. 2012. 1   Filter: A Simple Speed-Based Low-Pass Filter for Noisy Input in Interactive Systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) (CHI '12). Association for Computing Machinery, New York, NY, USA, 2527–2530. <https://doi.org/10.1145/2207676.2208639>
- [8] Mark Claypool and Kajal Claypool. 2009. Perspectives, Frame Rates and Resolutions: It’s All in the Game. In *Proceedings of the 4th International Conference on Foundations of Digital Games* (Orlando, Florida) (FDG '09). Association for Computing Machinery, New York, NY, USA, 42–49. <https://doi.org/10.1145/1536513.1536530>
- [9] Mark Claypool, Ragnhild Eg, and Kjetil Raaen. 2016. The Effects of Delay on Game Actions: Moving Target Selection with a Mouse. In *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts* (Austin, Texas, USA) (CHI PLAY Companion '16). Association for Computing Machinery, New York, NY, USA, 117–123. <https://doi.org/10.1145/2968120.2987743>
- [10] 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 the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) (CHI '15). Association for Computing Machinery, New York, NY, USA, 1827–1836. <https://doi.org/10.1145/2702123.2702300>
- [11] EliteOwnage 2013. *Pro Gaming Mouse Guide*. Retrieved July 05, 2021 from <https://eliteownage.com/mouseguide.html>
- [12] Sebastian Friston, Per Karlstr m, and Anthony Steed. 2015. The effects of low latency on pointing and steering tasks. *IEEE transactions on visualization and computer graphics* 22, 5 (2015), 1605–1615. <https://doi.org/10.1109/TVCG.2015.2446467>
- [13] ISO 4120:2004 2004. *Sensory analysis – Methodology – Triangle test*. Standard. International Organization for Standardization, Geneva, CH.
- [14] Zolja 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 the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 135–144. <https://doi.org/10.1145/2702123.2702432>
- [15] Benjamin F. Janzen and Robert J. Teather. 2014. Is 60 FPS Better than 30? The Impact of Frame Rate and Latency on Moving Target Selection. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1477–1482. <https://doi.org/10.1145/2559206.2581214>
- [16] Ricardo Jota, Albert 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 *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) (CHI '13). Association for Computing Machinery, New York, NY, USA, 2291–2300. <https://doi.org/10.1145/2470654.2481317>
- [17] I. Scott MacKenzie and Colin Ware. 1993. Lag as a Determinant of Human Performance in Interactive Systems. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems* (Amsterdam, The Netherlands) (CHI '93). Association for Computing Machinery, New York, NY, USA, 488–493. <https://doi.org/10.1145/169059.169431>
- [18] Mathieu Nancel, Stanislav Aranovskiy, Rosane Ushirobira, Denis Efimov, Sebastien Poulmane, Nicolas Roussel, and G ry Casiez. 2018. Next-Point Prediction for Direct Touch Using Finite-Time Derivative Estimation. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) (UIST '18). Association for Computing Machinery, New York, NY, USA, 793–807. <https://doi.org/10.1145/3242587.3242646>
- [19] Mathieu Nancel, Daniel Vogel, Bruno De Araujo, Ricardo Jota, and G ry Casiez. 2016. Next-Point Prediction Metrics for Perceived Spatial Errors. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo,

- Japan) (*UIST '16*). Association for Computing Machinery, New York, NY, USA, 271–285. <https://doi.org/10.1145/2984511.2984590>
- [20] Albert Ng, Julian Lepinski, Daniel Wigdor, Steven Sanders, and Paul Dietz. 2012. Designing for Low-Latency Direct-Touch Input. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (Cambridge, Massachusetts, USA) (*UIST '12*). Association for Computing Machinery, New York, NY, USA, 453–464. <https://doi.org/10.1145/2380116.2380174>
- [21] Eunji Park, Sangyoon Lee, Auejin Ham, Minyeop Choi, Sunjun Kim, and Byungjoo Lee. 2021. Secrets of Gosu: Understanding Physical Combat Skills of Professional Players in First-Person Shooters. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14. <https://doi.org/10.1145/3411764.3445217>
- [22] Andriy Pavlovych and Wolfgang Stuerzlinger. 2009. The Tradeoff between Spatial Jitter and Latency in Pointing Tasks. In *Proceedings of the 1st ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (Pittsburgh, PA, USA) (*EICS '09*). Association for Computing Machinery, New York, NY, USA, 187–196. <https://doi.org/10.1145/1570433.1570469>
- [23] ProSettings 2019. *What is mouse acceleration?* Retrieved July 05, 2021 from <https://prosettings.net/library/what-is-mouse-acceleration/>
- [24] Robert J Teather, Andriy Pavlovych, Wolfgang Stuerzlinger, and I Scott MacKenzie. 2009. Effects of tracking technology, latency, and spatial jitter on object movement. In *2009 IEEE Symposium on 3D User Interfaces*. 43–50. <https://doi.org/10.1109/3DUI.2009.4811204>
- [25] Ed Thorn. 2019. *League of Legends: Best mouse DPI and settings guide*. Retrieved July 05, 2021 from <https://www.metabomb.net/leagueoflegends/gameplay-guides/league-of-legends-best-mouse-dpi-and-settings-guide-2>
- [26] Thao Tran. 2021. *How to Turn off Mouse Acceleration, Everywhere*. Retrieved July 05, 2021 from <https://voltage.com/mouse-acceleration/>
- [27] Raphael Wimmer, Andreas Schmid, and Florian Bockes. 2019. On the Latency of USB-Connected Input Devices. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12. <https://doi.org/10.1145/3290605.3300650>
- [28] Jacob O. Wobbrock, Kristen Shinohara, and Alex Jansen. 2011. The Effects of Task Dimensionality, Endpoint Deviation, Throughput Calculation, and Experiment Design on Pointing Measures and Models. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) (*CHI '11*). Association for Computing Machinery, New York, NY, USA, 1639–1648. <https://doi.org/10.1145/1978942.1979181>