Evaluating Temporal Delays and Spatial Gaps in Overshoot-avoiding Mouse-pointing Operations

ABSTRACT
For hover-based UIs (e.g., pop-up windows) and scrollable UIs, we investigated mouse-pointing performance for users trying to avoid overshooting a target while aiming for it. Three experiments were conducted with a 1D pointing task in which overshooting was accepted (a) within a temporal delay, (b) via a spatial gap between the target and an unintended item, and (c) with both a delay and a gap. We found that, in general, movement times tended to increase with a shorter delay and a smaller gap if these parameters were independently tested. Therefore, Fitts’ law cannot accurately predict the movement times when various values of delay and/or gap are used. We found that 0.4 sec is a sub-optimal delay for densely arranged targets, but we found no optimal gap.

Author Keywords
Pointing; Fitts’ law; graphical user interfaces; human performance modeling.

CCS Concepts
+Human-centered computing → HCI theory, concepts and models; Pointing; Empirical studies in HCI;

INTRODUCTION
Background
In conventional studies on user performance in mouse-pointing tasks, users are instructed to point at and click an intended target as quickly and accurately as possible. The success or failure is judged by whether the mouse cursor falls within the target boundary when the mouse button is pressed, and the measured subjects (task results) are typically the movement time $MT$ and the error rate. Other measures of movement efficiencies have also been proposed by MacKenzie et al., such as Movement Variability [30]. In this paper, we revisit one of their proposed measures—Target Re-entry—for the case in which a cursor leaves the target and then enters it again. In particular, we focus on the case of overshooting, which is when the cursor passes through the target and then returns to it.

Such overshooting becomes problematic when, for example, a user is trying to select a target that pops up when the cursor hovers on a certain item, as shown in Figure 1a. For example, in Amazon Prime Video, when users move the cursor onto a movie thumbnail, a corresponding window pops up to show the details of Movie 3: the description, users’ rating, a Play button, etc. If users want to click on the Play button to start Movie 3, they should not largely overshoot it. If they do, the cursor leaves the pop-up window and hovers over the thumbnail of Movie 1, and unintentionally the pop-up window

1In some related papers, the term overshoot means that a click position falls beyond the target (e.g., [24]). In our paper, however, this term means that the cursor has crossed the target and has to move backward to return to it (e.g., [33]).
As shown in these examples, overshooting the target during aiming can be interpreted as an error operation in realistic UIs, although conventional pointing studies have considered only clicking outside the target as an error. The degree of care required to avoid overshooting depends on the task conditions. For example, when the Gap (or offset, margin) in Figure 1a is wide, the possibility of unintentionally closing the pop-up window of Movie 3 is small even if the user overshoots the target Play button. In the authors’ environment, the Gap is 36 pixels, but how does the user performance change if the Gap is reduced to 18 pixels?

Similarly, as implemented on many websites and OSs, pop-up windows that close by cursor-leaving can have a temporal delay (or timeout). Such a delay, typically configured during website development by means of the setTimeout (JavaScript) or delay (jQuery) functions, gives users a chance to notice if accidental overshooting has occurred and then return the cursor to the pop-up window. This is desirable because it frees users from having to concentrate too hard on precise mouse operations. For example, the delay to close the miniature windows in Figure 1b in our environment is approximately 400 msec, but we would have to point to the target window more carefully if it were 50 msec.

Research Question and Contribution Statement
It remains unclear how such gaps and delays help users. Possible drawbacks are, obviously, large gaps that take up a lot of screen space, and longer mouse cursor movements to point to intended items. For example, if the Gap in Figure 1a is very wide, when the user wants to view the details of Movie 1, the cursor movement distance to the thumbnail increases. Because GUI designers carefully manage the space, unnecessary empty space should be avoided if users are not helped by the gaps.

Moreover, if the delay to close a pop-up window is too long, the user has to wait a long time for the window to close. For example, in Figure 1a, when users want to view the description of Movie 2 that is behind the pop-up window of Movie 3, they have to remove the cursor from the thumbnail and wait for the delay to finish. Also, in Figure 1b, if users decide to click on a desktop icon behind the miniature windows, they have to remove the cursor from the taskbar icon and the miniature windows and then wait for the given delay.

In summary, short delays and small gaps have potentially negative effects on user performance (i.e., by enforcing careful operation). At the same time, unnecessarily long delays and wide gaps should also be avoided. Therefore, to clarify how such delays and gaps affect mouse-pointing performance, we conducted three controlled experiments. In the first and second experiments, we evaluated the effects of delays and gaps independently. The third experiment was to test the interaction between delay and gap. For example, the pop-up window in Amazon Prime Video immediately closes when the cursor leaves the window, but if designers configure a delay of 200 msec, users can overshoot the target and then hover over an unintended object (in Figure 1a, the thumbnail of Movie 1) for a short time without closing the pop-up window of Movie 3. This provides the chance to notice the overshooting and return to the intended target. Thus, users may be more relaxed when pointing to the Play button.

Our key contributions include:

(1) Conducting three experiments to evaluate the effects of delays and gaps in mouse-pointing tasks in which users have

2Dragging-and-dropping is also modeled by Fitts’ law [17, 23, 31].

3In Windows 10, manually clicking on the desktop does not close the miniature windows.
to avoid overshooting the target. The results showed that delays and gaps had different effects: the movement time $MT$ decreased with large gaps, while the effect of delays plateaued for 0.4 sec or longer.

(2) Evaluating Fitts’ law in all three experiments. Because delays and gaps significantly affected the $MT$, different intercepts and slopes should be used to accurately predict the $MT$ depending on these two factors.

RELATED WORK
The problematic operation discussed in this paper, namely, overshooting a target, becomes an issue when indirect pointing devices are used. In contrast, when direct input methods such as finger touching are used, typically the system cannot sense overshooting of a finger above the surface.

In Figure 1a, users have to avoid largely overshooting the target during a leftward horizontal movement, and they also must not deviate from the pop-up window on the y-axis. Such movements are called steering [1, 14, 36], but in this study we simplify our experiments by limiting these to 1D horizontal movements.

Effects of Delay in GUI Operations
The effects of delays and gaps, the main focus in this paper, have been discussed in the literature. The problem of delay (or lag) has been addressed in relation to remote operations [22] and virtual reality systems with non-responsive devices [32, 35, 43]. In these studies, lag means the time from user’s input to system’s response, and thus it is not directly relevant to our research focus.

Regarding conventional desktop environments with mouse operations, delays have often been mentioned in relation to drop-down or cascaded menus for opening sub-menus by cursor-hovering over a menu item. That is to say, increasing the delay can prevent unintended exposure of sub-menus, but a long delay can force the user to wait to open intended sub-menus after hovering over the parent menu item [11, 26, 40, 41]. While these studies focused on improving menu operations by, for example, reducing the movement time and the error rate, the effects of delay on the menu pointing time have not been modeled. Hence, we cannot refer to or modify the models for menu selection times. Bailly et al.’s recent survey [3] provides a thorough review of visual menu selection techniques. However, these studies do not directly answer our questions about how a delay helps users’ pointing operations with avoiding overshooting, even though such operations are required in typical PC tasks (Figures 1 and 2).

Yamanaka proposed a refined version of steering law [1] with a temporal-delay term to accept errors (deviation from a path) [49]. A difference in his work is that the necessary precision in path-steering tasks is higher than pointing; the cursor must not deviate from a path, and thus a delay helps throughout the path. As a result, the $MT$ was reduced by 54% at most in steering [49] but only by 17% at most in our experiments. Thus, designers should accept that the $MT$ in pointing cannot be largely reduced even if a long delay is set. As a series of work on modeling the $MT$ with delay, our work would be beneficial to model a targeted-steering task [13, 28, 38, 42], e.g., clicking a target after steering a path, such as menu selection. For example, in Figure 1, users must not deviate from the pop-up window on the y-axis and then click the target. Our work opens up new research on such common tasks.

Effects of Gaps in Target Arrangements
If GUI items are densely arranged (i.e., small Gap), users tend to carefully point to the intended target. The effects of other unwanted items, hereafter distractors, have been studied. The area cursor [24] and its variations [10, 18] are typical examples affected by distractors, as the cursor size cannot be expanded among a dense group of distractors. A target-aware cursor jumping technique [19], target-aware gain-changing techniques [5, 46], and use of multiple cursors [27] are other examples affected by the distractor density. In the experiments undertaken for these studies, overshooting the target while aiming was permitted, as in the conventional Fitts’ law task.

To model the density effect on these techniques, Blanch and Ortega introduced the index of sparseness of potential targets into Fitts’ law [6]. Also in their model, overshooting while aiming was permitted, while in our experiments, such an operation is considered an error. Thus, the index of target sparseness could not be equivalently applied to our intended tasks in Figures 1 and 2.

Performance Models of Pointing Tasks
For predicting user performance in aiming tasks, a promising model is Fitts’ law [15] in the Shannon formulation [29, 39]:

$$MT = a + b \log_2 \left( \frac{A}{W + 1} \right)$$  \hspace{1cm} (1)

where $MT$ is the time to point to the target, $A$ is the distance to the target, and $W$ is its size. $a$ and $b$ are empirically determined constants. The logarithmic term is called the index of difficulty ($ID$):

$$ID = \log_2 \left( \frac{A}{W + 1} \right)$$  \hspace{1cm} (2)

Fitts’ law with nominal $A$ and $W$ values shows a good fit to error-free data [51].

Correcting the data when targets are missed has been discussed in previous works. In HCI, replacing $W$ in Equation 2 with the effective width ($W_e$) is recommended [29, 39]. Here, $W_e$ is $4.133 \times \sigma$, where $\sigma$ is the standard deviation of the click positions.

As shown by Wright and Lee [47], however, the effective width method is not appropriate for our purpose. This is because, when designers set a target size to a new value (e.g., $W = 30$ pixels) after conducting a user study, they can estimate the $MT$ for selecting the target by using $W$ but cannot use $W_e$ without conducting a new user study. Thus, for predicting an average operation time when designing a GUI, using the nominal $ID$ value is more appropriate, as was also mentioned by Zhai et al. [51]. More seriously, if many click positions fall outside a target, we obtain a larger $W_e$ value than the nominal $W$; e.g., $W_e = 34$ pixels for a target of $W = 30$ pixels. This conflicts with our rule that the user must not overshoot a given target area when there is no gap. Thus, $W_e$ was not used in this study, and in our data analysis we use nominal values for Fitts’ law.
Effects of Delay in Fitts’ Law Tasks

System latency or lag is defined as “the time interval between a user’s action and the system’s response,” (also called an end-to-end lag [12]). In a mouse-pointing task, the lag results from several factors such as the mouse’s USB driver, the programming language, and the GPU (the factors are well reviewed in [2]). To capture the negative effect of lag, Hoffmann [20] and MacKenzie and Ware [32] independently proposed the same model modified from Fitts’ law and found that $MT$ increases as the cursor’s latency from physical mouse movements increases. This model is explained in our Supplementary Material.

In contrast, in our study, we did not add any such artificial lag from mouse positioning to cursor movement. Our intentional delay was added to close a pop-up window when the cursor leaves it. Therefore, the lag model [20, 32] cannot be straightforwardly applied here.

Intuitively, we assume that a longer delay reduces the negative effect on user performance. In addition, if designers make a larger gap between the target and a distractor (say, 100 pixels rather than 20 pixels), the user would move the mouse more quickly, as the possibility to leave the pop-up window decreases. Empirical evidence on these effects will thus contribute to a better understanding of fundamental human motor behaviors and GUI designs.

**EXPERIMENT 1: EFFECTS OF DELAY**

We concurrently conducted Experiments 1 and 2 with the same participants on the same day. In Experiment 1, we evaluated the effects of a temporal delay in leaving then closing pop-up windows ($T_{delay}$), and in Experiment 2, we evaluated the effects of the Gap. Each experiment took 20 min to complete. The order of the two studies was balanced among the 12 participants. This allowed us to compare the results in Experiments 1 and 2. For example, if $T_{delay}$ significantly affected the $MT$ but Gap did not, we can assume it is not due to any difference of the participant group. An experiment with 12 participants is common in the HCI field (particularly in the CHI literature [7]).

**Participants**

Twelve participants were recruited from a local university (all men; ages: $M = 22.5$, $SD = 1.19$ years). All had normal or corrected-to-normal vision, were right-handed, and were familiar with mouse operations. Three of them were daily mouse users. Each participant received 45 USD for his time (~40 min total for Experiments 1 and 2).

**Apparatus**

The PC we used was a Sony Vaio Z (Core i7-5557U, 3.10 GHz, 4 cores; 16-GB RAM; Windows 10). The display was manufactured by Dell (2407WFPb: 24-inch diagonal, 1920 × 1200 pixel resolution, 518.4 × 324.0 mm display area, 3.70 pixels/mm; 16-ms response time; connected by an HDMI-to-DVI cable), and its refresh rate was set to 60 Hz. The input device was an iBuffalo optical mouse (BSMBU05: blue LED, 81.6 g, 1000 dpi; 1.5-m cable). We used a large mousepad (43 cm × 29 cm). The experimental system was implemented with Hot Soup Processor 3.4 and used in full-screen mode. The system read and processed input approximately 1000 times per second.

The mouse-cursor speed was set as the default in the OS, i.e., the control-display gain was in the middle of the slider. Pointer acceleration, denoted as the Enhance pointer precision setting in Windows 10, was enabled to allow the participants to perform mouse operations with higher ecological validity [9]. Using pointer acceleration does not violate a Fitts task (e.g., [48]), and it is consistent with the settings of a consumer OS such as Windows or macOS.

The latency was measured with a Casio Exilim EX-ZR4000WE camera at 1000 fps. The mouse was hit with a hard object at high speed, and the number of frames from when the mouse stopped to when the cursor stopped was counted. We repeated this action 30 times, and the average latency was 57.9 msec ($SD = 11.2$). This is in the range of typical mouse-display latencies of approximately 55 to 82 ms [8]. Therefore, we assume that the latency of our experimental system did not have a significant negative effect on user performance.

**Task**

We used discrete pointing tasks, as shown in Figure 3a. First, participants clicked on the green start bar labeled “1”, and then they clicked on the blue target bar labeled “2”. If the cursor (a) overshot the target, (b) hovered over the pink distractor, and (c) had a hovering time above a given duration $T_{delay}$, then the trial was considered an error of closing the pop-up window $ER_{close}$. The $ER_{close}$ was differentiated from a typical error in pointing, or $ER_{click}$, which simply occurred when the click position was outside the target.

The participants were instructed to select the target as quickly as possible and to avoid making an error ($ER_{close}$ or $ER_{click}$). It did not matter if the cursor returned to the target area within the given $T_{delay}$. In Experiment 1, there was no gap between the target and distractor ($Gap = 0$ pixel).

When an $ER_{close}$ occurred, the pink distractor turned red, and a “friction” sound was played. When an $ER_{click}$ occurred, a beep sound was played. Even if an erroneous operation was performed, the participants had to immediately aim for the target again; the task was not restarted from the beginning.

**Design and Procedure**

We tested six delay values ($T_{delay} = 0, 0.1, 0.2, 0.4, 0.8$, and $\infty$). Participants were not permitted any overshooting under the $T_{delay} = 0$ sec condition. The $T_{delay} = \infty$ condition, under which the participants did not have to worry about overshooting, was included to determine the baseline performance as in conventional Fitts tasks. The middle four values were selected...
on the basis of typical human reaction times. Because the time to correct hand movements according to visual feedback is longer than 200 msec [37]—approximately 260 [25] or 290 msec [34]—the participants could presumably return the cursor within ~300 msec (including the system latency of 57.9 msec) after they noticed overshooting. Therefore, to observe the effects of delay, we set the $T_{\text{delay}}$ values to range from less than to sufficiently longer than human reaction time.

Two target distances ($A = 250$ and 600 pixels) and three widths ($W = 25, 45$, and 75 pixels) were tested. When the ID is smaller than approximately 3 or 4 bits, participants perform ballistic (feedforward) pointing motions [16, 21], such as aiming for a large or close target. Pointing to a close target is likely to occur in daily PC work, as shown in Figure 1, and thus we set ID to range from 2.7 to 5.6 bits to cover ballistic to visually controlled pointing motions. Note that the ID here is the original formulation by Fitts: $ID = \log_2(2A/W)$ [15]. Because our main focus is on $T_{\text{delay}}$ and Gap, the numbers of $A$ and $W$ were relatively small.

One block consisted of a random order of $3A \times 2W \times 10$ repetitions = 60 trials with a fixed $T_{\text{delay}}$ value. The first repetition was considered practice. In addition, before each block, the participants used an exercise system (described below) to learn the delay of the next block, except for the $T_{\text{delay}} = 0$ sec and $\infty$ conditions. The order of the six $T_{\text{delay}}$ values was balanced among the 12 participants. The movement direction was always to the right. In total, we recorded $3A \times 2W \times 9$ repetitions $\times 6T_{\text{delay}} \times 12$ participants = 3888 data points.

Exercise to Learn a Given Delay
As shown in Figure 3b, the participants moved the cursor to the right pink area, and then a time measurement began. The goal was to return the cursor to the left gray area before the given $T_{\text{delay}}$. If the measured time was over the $T_{\text{delay}}$, the pink area turned red and a friction sound was played. The participants repeatedly performed this task and were expected to learn how to immediately return the cursor after entering the pink area. The session finished when a participant felt that he had sufficiently learned the $T_{\text{delay}}$ value. The time required was typically 30–40 sec (~15 or 20 trials).

This exercise session allowed us to simulate a situation in which the participants were already familiar with the delay of a certain GUI. If we had not used this system, we would have been simulating participants using a GUI for the first time and learning the delay at every trial, but our purpose here was not to observe such a learning effect.

Results
We removed one spatial outlier data point (0.026%) whose movement distance was less than $A/2$ or whose click position was more than $2W$ from the target center [4]. The remaining 3888 data points were analyzed using repeated-measures ANOVA with Bonferroni correction as the p-value adjustment method. The main effects of $T_{\text{delay}}$ on the $MT$, $ER_{\text{click}}$ rate, and $ER_{\text{close}}$ rate are shown in Figure 4.

Throughout the three experiments reported in this paper, Mauchly’s sphericity test was used when we ran $F$ statistic. Because all the results on the sphericity assumption for $MT$, $ER_{\text{click}}$ rate, and $ER_{\text{close}}$ rate showed $p > 0.05$, all degrees of freedom of $F$ statistic are reported as integers without using any correction methods (e.g., Greenhouse-Geisser).

Movement Time ($MT$)
Again, when analyzing the $MT$ results and Fitts’ law fitness, we used error-free data. We found significant main effects for $T_{\text{delay}}$ ($F_{5,55} = 12.45$, $p < 0.001$, $\eta_p^2 = 0.531$), $A$ ($F_{1,11} = 139.1$, $p < 0.001$, $\eta_p^2 = 0.927$), and $W$ ($F_{2,22} = 296.7$, $p < 0.001$, $\eta_p^2 = 0.964$). Significant interactions were found for $T_{\text{delay}} \times A$ ($F_{5,55} = 2.504$, $p < 0.05$, $\eta_p^2 = 0.185$) and $A \times W$ ($F_{2,22} = 10.92$, $p < 0.01$, $\eta_p^2 = 0.498$).

$ER_{\text{click}}$ rate
Regarding conventional pointing misses, we observed 181 $ER_{\text{click}}$ trials (46.6%). We did not find significant main effects for $T_{\text{delay}}$ ($F_{5,55} = 0.8870$, $p = 0.496$, $\eta_p^2 = 0.075$), $A$ ($F_{1,11} = 0.9530$, $p = 0.350$, $\eta_p^2 = 0.080$), or $W$ ($F_{2,22} = 1.932$, $p = 0.169$, $\eta_p^2 = 0.149$). Moreover, no significant interaction was found for $T_{\text{delay}} \times A$, $T_{\text{delay}} \times W$, or $A \times W$ ($p > 0.05$ for all).

$ER_{\text{close}}$ rate
We observed 167 $ER_{\text{close}}$ trials (43.0%) and found significant main effects for $T_{\text{delay}}$ ($F_{5,55} = 12.42$, $p < 0.001$, $\eta_p^2 = 0.530$) and $W$ ($F_{2,22} = 14.01$, $p < 0.001$, $\eta_p^2 = 0.560$), but not for $A$ ($F_{1,11} = 3.517$, $p = 0.088$, $\eta_p^2 = 0.242$). Significant interactions were found for $T_{\text{delay}} \times A$ ($F_{5,55} = 3.807$, $\eta_p^2 = 0.257$), $T_{\text{delay}} \times W$ ($F_{5,55} = 3.917$, $\eta_p^2 = 0.263$), and $A \times W$ ($F_{2,22} = 5.551$, $\eta_p^2 = 0.335$). Note that $ER_{\text{click}}$ and $ER_{\text{close}}$ could occur concurrently in one trial (inclusive). The number of error-free trials was 3571, while 316 data points (8.13%) were removed because of errors found when analyzing the $MT$.

Model Fitting
Fitts’ law showed $R^2 > 0.99$ in each case for all $T_{\text{delay}}$ conditions by using $N = 6 (2A \times 3W)$ data points as shown in Figure 5. This means that the $MT$ can be accurately predicted if we use a single $T_{\text{delay}}$ value. However, if several $T_{\text{delay}}$ conditions are mixed, the prediction accuracy was comparatively low ($R^2 = 0.882$).

Speed Profiles
Next, we investigated how the $T_{\text{delay}}$ affected pointing behaviors during aiming. Figure 6 shows the speed profiles for each
Another important finding is that this benefit due to a long $T_{delay}$ was clearly higher than the other three conditions. This result indicates that $T_{delay}$ helped the participants to accelerate the pointing speed, but if $T_{delay}$ was shorter than human reaction time, they could not effectively take advantage of such an overshoot-accepting condition.

Another important finding is that this benefit due to a long $T_{delay}$ increased the peak speeds, and a higher speed was maintained until entering the target area compared to the no-delay condition. Hence, we assume that the participants were more relaxed by a longer $T_{delay}$ not only during the final cursor positioning phase onto the target, but also throughout the feedback-loop controlling phase.

**Discussion of Experiment 1**

Regardless of the degree of care in avoiding overshooting due to the $T_{delay}$, Fitts’ law showed $R^2 > 0.99$, while the MT showed significant differences. Because we tested $T_{delay}$ ranging from 0 to $\infty$ sec, this high fitness will be observed for untested $T_{delay}$ values such as 1 or 2 sec. This result can help designers predict an MT value for a given set of $A$ and $W$ if a $T_{delay}$ value is fixed. However, if designers want to use a new $T_{delay}$ value, they will have to conduct another user study, as Fitts’ law for $N = 36$ data points showed $R^2 = 0.88$.

We had assumed that a longer $T_{delay}$ value would enable the participants to move the cursor more quickly. Yet, we found no significant difference in the MT for $T_{delay}$ of 0.1 sec or longer. In addition, $T_{delay}$ showed no main effect on the $ER_{click}$ rate, and pairwise tests of the $ER_{close}$ rate showed no significant differences for $T_{delay}$ of 0.4 sec or longer. Therefore, in summary, the upper $T_{delay}$ value to remove the negative effects of the overshoot-avoiding condition was 0.4 sec. For $T_{delay}$ of 0.8 sec or longer, no advantages for MT or $ER_{close}$ rate would be gained. For $T_{delay}$ of 0.2 sec or shorter, negative effects to increase MT and $ER_{close}$ rate would be observed.

**EXPERIMENT 2: EFFECTS OF GAP**

**Design and Procedure**

Experiment 2 was concurrently performed with Experiment 1 using the same participants and the same apparatus. We tested six Gap values (0, 8, 24, 72, 216, and $\infty$ pixels). This task simulated that the Gap area was still inside a pop-up window (as in Figure 1a), and the pink distractor area was outside it. The participants were not permitted any overshooting under the Gap = 0 pixels condition. The conditions for $A$, $W$, and the movement direction were the same as in Experiment 1. Because $T_{delay}$ was fixed to 0 sec in Experiment 2, entering the pink distractor was not permitted, and the exercise system described for Experiment 1 was not used.

One block consisted of a random order of $3A \times 2W \times 6Gap = 36$ trials. One block for practice and then nine blocks for data collection were performed. In total, we recorded $3A \times 2W \times 6Gap \times 9$ repetitions $\times 12$ participants = 3888 data points.

**Results**

**Movement Time (MT)**

The main effects of Gap are shown in Figure 7. After removing two outlier data points (0.051%), we found significant main effects for $Gap$ ($F_{5,55} = 14.98, p < 0.001, \eta^2_p = 0.577$), $A$ ($F_{1,11} = 194.3, p < 0.001, \eta^2_p = 0.946$), and $W$ ($F_{2,22} = 147.6, p < 0.001, \eta^2_p = 0.931$). A significant interaction was found for $A \times W$ ($F_{2,22} = 13.09, p < 0.001, \eta^2_p = 0.543$).

**$ER_{click}$ rate**

We observed 208 $ER_{click}$ trials (5.35%). We found significant main effects for $Gap$ ($F_{5,55} = 3.386, p < 0.05, \eta^2_p = 0.235$) and $W$ ($F_{2,22} = 7.916, p < 0.01, \eta^2_p = 0.418$), but not for $A$ ($F_{1,11} = 0.360, p = 0.561, \eta^2_p = 0.032$). Significant interactions were found for $Gap \times A$ ($F_{5,55} = 2.826, p < 0.05, \eta^2_p = 0.204$) and $Gap \times W$ ($F_{10,110} = 3.013, p < 0.01, \eta^2_p = 0.215$).

**$ER_{close}$ rate**

We observed 131 $ER_{close}$ trials (3.37%). We found significant main effects for $Gap$ ($F_{5,55} = 19.31, p < 0.001, \eta^2_p = 0.637$) and $W$ ($F_{2,22} = 6.753, p < 0.01, \eta^2_p = 0.380$), but not for $A$ ($F_{1,11} = 4.181, p = 0.066, \eta^2_p = 0.275$). Significant interactions were found for $Gap \times A$ ($F_{5,55} = 2.553, p < 0.05, \eta^2_p = 0.188$) and $Gap \times W$ ($F_{10,110} = 2.349, p < 0.05, \eta^2_p = 0.176$).
The number of error-free trials was 3569, while 317 data points (8.16%) were removed because of errors found when analyzing the MT.

Model Fitting
Fits’ law showed $R^2 > 0.98$ for all Gap conditions by using $N = 6$ data points, as shown in Figure 8. When we did not separate the Gap conditions, the fit using $N = 36$ data points was $R^2 = 0.940$.

Speed Profiles
Figure 6 shows the speed profiles for each Gap under $A = 600$ and $W = 25$ pixels. Compared with the speed profiles in Experiment 1, we could not see clear differences between Gap values, while wider Gap values (216 and $\infty$ pixels) seemed to slightly increase the speed. This resulted in slight improvements in MT due to Gap values.

Discussion of Experiment 2
The results of Experiment 2 were quite different from those of Experiment 1. As shown in Figure 7a, MT decreased as Gap increased in Experiment 2, while the increase in MT caused by $T_{delay}$ plateaued for $T_{delay} \geq 0.1 \text{ sec}$ in Experiment 1. The model fitness without separating the Gap values was $R^2 = 0.94$, so designers can predict the average MT under a given condition regardless of the Gap with a certain degree of accuracy. Because we used the same participants in both Experiments 1 and 2 and the order of the two studies was balanced, we conclude that the resultant differences probably stem from whether the “allowance” of overshooting was given temporally ($T_{delay}$) or spatially (Gap).

EXPERIMENT 3: INTERACTIONS OF DELAY AND GAP
In Experiment 2, we found that the MT decreased as the Gap increased (Figure 7a). If the $T_{delay}$ and Gap were specified concurrently, however, the result might be different. For example, when the $T_{delay}$ is sufficiently long (e.g., 0.4 sec), there is little concern about invoking $ER_{close}$ (Figure 4c), and hence, the positive effect of a large Gap should disappear. To confirm such potential interactions between the $T_{delay}$ and Gap, we conducted Experiment 3, which took 25 to 30 min per participant.

Apparatus and Participants
Experiment 3 was conducted two weeks after Experiments 1 and 2. The same apparatus was used. Twelve participants were again recruited from a local university (four women, eight men; ages: $M = 22.2, SD = 1.72$ years). All had normal or corrected-to-normal vision and were right-handed. Two of them were daily mouse users. Three of them had also participated in Experiments 1 and 2. Each participant received 23 USD in compensation.

Design and Procedure
Because the goal of Experiment 3 was to observe potential interactions between the $T_{delay}$ and Gap, we did not evaluate any baseline performance (i.e., the cases of $T_{delay} = \infty$ sec and $Gap = \infty$ pixels). If we had included $T_{delay} = \infty$, there would be no concern about overshooting regardless of the Gap values, and likewise for $Gap = \infty$. Also, because $T_{delay} = 0.8$ sec showed no significant difference from $T_{delay} = \infty$ for any results ($MT, ER_{click}$, and $ER_{close}$; Figure 4), we used $T_{delay} = 0.4$ sec as the upper value. Thus, we reused four $T_{delay}$ values (0, 0.1, 0.2, and 0.4 sec) and five Gap values (0, 8, 24, 72, and 216 pixels). The $A$ and $W$ values were the same as in Experiment 1, and the movement direction was always to the right.

One block consisted of a random order of $3A \times 2W \times 5Gap \times 4$ repetitions = 120 trials with a fixed $T_{delay}$ value. Before each block, the participants used the exercise system as in Experiment 1, and then ten trials randomly selected from the 30 ($3A \times 2W \times 5Gap$) conditions were performed as practice. In total, we recorded $3A \times 2W \times 5Gap \times 4$ repetitions $\times 4T_{delay} \times 12$ participants = 5760 data points.

Results
Movement Time (MT)
The main effects of $T_{delay}$ and Gap are shown in Figure 10. After removing two outlier data points (0.035%), we found significant main effects for Gap ($F_{4,44} = 20.71, p < 0.001, \eta^2_p = 0.653$), $A (F_{1,11} = 188.48, p < 0.001, \eta^2_p = 0.945$), and $W (F_{2,22} = 55.12, p < 0.001, \eta^2_p = 0.981$), but not for $T_{delay}$.
We observed 257 trials (4.46%). We found significant main effects for \( T_{delay} \) (\( F_{3,33} = 4.967, \ p < 0.01, \ \eta_p^2 = 0.331, \)) and \( Gap \) (\( F_{4,44} = 4.770, \ p < 0.01, \ \eta_p^2 = 0.302, \)) and \( W \) (\( F_{2,22} = 4.633, \ p < 0.05, \ \eta_p^2 = 0.296, \)) but not for \( A \) (\( F_{1,11} = 0.082, \ p = 0.780, \ \eta_p^2 = 0.007). \) No significant interaction was found (\( p > 0.05). \)

\( ER_{close} \) Rate
We observed 145 \( ER_{close} \) trials (2.52%). We found significant main effects for \( T_{delay} \) (\( F_{3,33} = 22.03, \ p < 0.001, \ \eta_p^2 = 0.667), \) \( Gap \) (\( F_{4,44} = 20.82, \ p < 0.001, \ \eta_p^2 = 0.654, \)) and \( W \) (\( F_{2,22} = 6.474, \ p < 0.01, \ \eta_p^2 = 0.371, \)) but not for \( A \) (\( F_{1,11} = 0.171, \ p = 0.687, \ \eta_p^2 = 0.015). \) A significant interaction was found for \( T_{delay} \times Gap \) (\( F_{12,132} = 3.127, \ p < 0.01, \ \eta_p^2 = 0.221). \) Figure 11 shows this interaction. The number of error-free trials was 5389, while 369 data points (6.41%) were removed because of errors found when analyzing the \( MT. \)

Model Fitting
The \( R^2 \) values of the law for each \( T_{delay} \times Gap \) condition by using \( N = 6 \) (2A 3W) data points ranged from 0.914 to 0.999. When we did not separate the 4 \( T_{delay} \) and 5 \( Gap \) conditions, the fit using \( N = 120 \) data points was \( R^2 = 0.930. \)

Discussion of Experiment 3
Differently from the result of Experiment 1, \( T_{delay} \) showed no main effect on the \( MT. \) In contrast, a wider \( Gap \) significantly decreased the \( MT, \) which was consistent with Experiment 2. Still, the difference in \( MT \) with the minimum and maximum \( Gap \) values was small (696 – 646 = 50 msec). In short, our assumption that the \( T_{delay} \) and \( Gap \) would have a significant interaction for \( MT \) was rejected. Therefore, to model the \( MT \) data obtained in Experiment 3, it would be sufficient to account for \( Gap \) without \( T_{delay} \).

In contrast to the \( MT \) results, our assumption on the interaction of \( T_{delay} \times Gap \) on the \( ER_{close} \) rate was confirmed. This result shows that, if the effect to reduce the \( ER_{close} \) already has been achieved by either \( T_{delay} \) or \( Gap, \) the other parameter’s effect would be limited. For example, for \( Gap = 24 \) pixels in Figure 11, the \( ER_{close} \) rates are already small (<5%), and thus the \( ER_{close} \) rates did not significantly change even if \( T_{delay} \) increased. This finding could not have been observed if we had tested only the effect of \( T_{delay}; \) as shown in Figure 10e, the \( ER_{close} \) rate of \( T_{delay} = 0.4 \) sec was significantly different from the other three values.

GENERAL DISCUSSION
Experimental Design Issue of Learning Effects
For fair comparison, we checked the learning effects on \( MT, \) \( ER_{click} \) rate, and \( ER_{close} \) rate in overshoot-prohibited conditions (\( T_{delay} = 0 \) msec and \( Gap = 0 \) pixels). The effects of order (Experiments 1 vs. 2) are not significant for \( MT \) (a pair-wised two-tailed t-test shows \( p = 0.098) \) and \( ER_{click} \) and \( ER_{close} \) rates (Wilcoxon signed-rank tests showed \( p = 0.86 \) and \( p = 0.24, \) respectively). For the three participants joining in Experiments 1–3, the effects of order (Experiments 1 vs. 2 vs. 3) on \( MT \)’s and \( ER_{close} \) rate are not significant (repeated-measures ANOVA showed \( p = 0.48 \) and 0.86, respectively). The \( ER_{click} \) rate shows \( p < 0.05 \) but pair-wise tests show no significant differences in all pairs. This result rejects our concerns that learning effects change our conclusions.

Participants’ Strategy
While the effects of the \( T_{delay} \) and \( Gap \) on \( MT, \) \( ER_{click} \) rate, and \( ER_{close} \) rate were independently observed in Experiments 1 and 2, the concurrent interaction effects of these factors were observed only for the \( ER_{close} \) rate in Experiment 3. According to oral interviews, in Experiment 1, 11 of the 12 participants stated that they changed the mouse movement speed depending on the \( T_{delay}. \) Similarly, in Experiment 2, eight participants stated that they changed the speed depending on the \( Gap. \) In contrast, in Experiment 3, six and seven participants stated that they changed the speed depending on the \( T_{delay} \) and \( Gap, \) respectively (inclusive). The smaller number of participants who changed the speed may be one reason that there was no significant difference for \( T_{delay} \) on \( MT \) in Experiment 3.
Hence, not many users purposely adjusted their movement speed depending on the given parameters of $T_{delay}$ and Gap in Experiment 3. Rather, the priority that affected user performance was strongly on the nominal ID. In a real GUI, however, both delays and gaps can be independently designed. Further theoretical development of the model will help GUI designers, but for now, predicting the MT by using the baseline Fitts’ law model is a suboptimal approach.

In all three experiments, we measured user performance under conditions where the participants knew the $T_{delay}$ and the target and Gap areas were explicitly drawn. In Netflix, for example, hovering the cursor over a movie thumbnail reveals a Play button and detailed descriptions. Similarly to Amazon Prime Video, the circular Play button has a Gap from the neighboring movie thumbnail visually. However, that button has a quite large area that receives mouse-click events, and there is no Gap from the neighboring thumbnail in the motor space. In such a case, i.e., where a target has visual and motor spaces of different sizes, user performance would degraded [44, 45], but our findings are limited to a condition of the same visual and motor sizes.

Suboptimal Design Recommendation

The results of Experiment 1 show that the MT was significantly worse for $T_{delay} = 0$ sec (Figure 4a), but the negative effect plateaued for $T_{delay} \geq 0.1$ sec. In addition, the $T_{delay}$ did not significantly affect the $ER_{click}$ rate (Figure 4b), and $T_{delay} \geq 0.4$ sec did not show a significantly worse $ER_{close}$ rate (Figure 4c). Therefore, when there is no gap beyond the target and distractor, using a 0.4-sec delay to close a pop-up window seems a suboptimal choice. This also justifies the delay configuration for miniature windows shown in Figure 1b.

Regarding the Gap, our results showed that greater gap values were better in terms of decreasing the MT and errors in general (Figures 7 and 10). On the other hand, a wider gap directly occupies a large space. Hence, our results suggest no clear optimal value for the Gap that minimizes wasted spaces while maintaining user performance.

Based on the results of Experiment 3, $T_{delay}$ and Gap had a significant interaction on $ER_{close}$ rate (Figure 11), and thus it prevents us from recommending specific values of these parameters. In summary, from the results of Experiments 1–3, we conclude that $T_{delay} = 0.4$ sec is a suboptimal value for gap-absent target arrangements, and that there is no clear evidence of an optimal Gap value. In addition, there remains a possibility that the actual suboptimal delay is an untested value around 0.4 sec, such as 0.3 or 0.5 sec.

Limitations and Future Work

Our results cannot be generalized to other conditions besides our experimental setup. For example, cursor responsiveness is affected by mouse-to-cursor transfer functions and by differences between pointing devices such as mouse vs. touchpad [9]. The responsiveness (or transmission lag) significantly affects the pointing performance and Fitts’ law fitness [20, 32]. A fixed movement direction is also a limitation, as mouse-pointing performance varies depending on directions [50, 52]. Hence, we do not claim that we have found an optimal delay or gap, or that we have found an optimal menu design.

While our tasks were limited to 1D pointing, we are also interested in conducting experiments with a target and distractor of finite heights, which is more realistic for current GUIs (as shown in Figure 1). From this viewpoint, we feel that our 1D tasks were somewhat simple compared to real GUIs. Even so, the three experiments provide good motivation for further studies on the topic of overshoot-avoiding pointing performance.

Our future work includes deriving a model in a more theoretical way. The lag model [20, 32] assumes that every loop in corrective movements slows down, and this was empirically validated via cursor trajectories with timestamps [22]. In our study, we observed a similar effect of $T_{delay}$ on the cursor speed profiles in Experiment 1 (Figure 6). However, such speed changing was not clearly observed in Experiment 2 (Figure 9). Thus, carefully analyzing the principles involved in changing the movement speed will provide better models to explain both the $T_{delay}$ and Gap effects.

An unclear point regarding GUI designs is that, although 0.4 sec seems the upper value for ignoring the negative effects of the $T_{delay}$ in Experiment 1, it is possible that such a delay negatively affects users’ perceptions and feelings if users want to view the items behind the pop-up window. In addition to the quantitative experiments conducted in this study, subjective evaluation is needed in order to judge whether this value is preferred by users.

CONCLUSION

We conducted three mouse-pointing experiments in which users had to avoid overshooting a target. User performance was significantly degraded by a shorter delay for acceptance of overshooting and a smaller gap between the target and distractor. Fitts’ law held if we run regression expressions for each value of delay and gap, but a more theoretical derivation will be required to capture the effects on MT. We found that 0.4 sec is a suboptimal delay for densely arranged targets, but we found no optimal gap.

REFERENCES


