

# Mouse Cursor Movements towards Targets on the Same Screen Edge

Shota Yamanaka\*

Yahoo Japan Corporation

## ABSTRACT

Buttons and icons on screen edges can be selected in a shorter time than those in the central area because the mouse cursor stops due to the impenetrable borderline. However, we have concerns regarding such edge targets, in particular, pointing to an edge target from another edge target on the same edge. For example, users would move the mouse toward outside the screen; thus, the *virtual travel distance* of the cursor including off-screen movements would be longer. In this study, we empirically confirmed that users exhibit such “pushing-edge” behavior, and 3% of cursor movements are wasted in off-screen movements. We also report how current user-performance models (variations of Fitts’ law) can capture such pointing motions between targets on the same edge. The results show that the baseline model (Shannon formula) shows a reasonably high fit ( $R^2 = 0.959$ ), and bivariate pointing models show higher fitness ( $R^2 = 0.966$  at most).

**Keywords:** Graphical user interfaces, mouse pointing, edge-targets, screen edge, Fitts’ law.

**Index Terms:** H.5.2. [Information interfaces and presentation]: User Interfaces - Input devices and strategies

## 1 INTRODUCTION

Buttons or icons on edges of PC screens can be pointed to in a shorter time than those in the central area. This advantage of *edge targets* has been theoretically and empirically demonstrated [3]. Hence, as a general guideline, targets that will be often used should be arranged on screen edges or corners [18]. While research on pointing actions for targets in the central area is more common, edge targets are also widely used in modern GUIs.

In this paper, we investigate the characteristics in mouse-pointing operations particularly for pointing to an edge target from another edge target on the same screen edge. For example, users switch applications by clicking the *task bar* on Windows OS, and open sub-menus by hovering on the *menu bar* on Mac OS. Edge targets would also appear at the left and right edges of the screen, e.g., showing a certain stream by clicking a menu on the left-screen edge of TweetDeck. The *task bar* of Windows and *Dock* of Mac can also be repositioned on the left and right edges by users.

The basic motivation to conduct our experiment comes from our observations of some users’ mouse operations. When they switched the tabs of Google Chrome to take a look at other websites, they moved the mouse a little obliquely upwards rather than horizontally, as shown in Figure 1, with pivoting their wrists. We asked them the reason they moved the mouse in such a manner, and some users answered that they wanted to avoid the risk of clicking outside the intended tab by moving the cursor

unintentionally downwards a little. We believe that this explanation is reasonable because when humans horizontally move their hand, a little “noise” which vertically moves the hand is difficult to be avoided [33]. To keep the cursor inside the range of Chrome tabs on the y-axis, moving the mouse upwards on purpose seems to be a beneficial strategy. We refer to this as “pushing-edge” behavior.

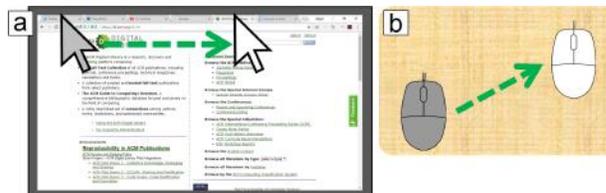


Figure 1: Mouse operation to switch tabs of Chrome. (a) While the cursor moves horizontally along the *North* (top) edge of the screen, (b) users tend to move the mouse a little obliquely upwards.

Although pushing-edge behavior would be safe for pointing to an edge target, there are several concerns, e.g., when the cursor stops at the *North* (top) edge of the screen, hand movements along the vertical axis are not reflected on the screen. Thus, vertical movements could be regarded as “wasted” effort. In addition, such extra hand movements would increase the movement time (*MT*) of the pointing task. We also wonder if users would exhibit the same behavior under other conditions, e.g., the other three edges (*South*, *West*, and *East*), or different target sizes. Moreover, we observed the pushing-edge behavior exhibited by only limited users. Hence, we would like to empirically test whether other users would generally exhibit this behavior.

Our main contributions are as follows:

- 1) We empirically show that users tend to exhibit pushing-edge behavior when pointing to edge targets on the same screen edge. This results in 3% of cursor movements that are not reflected on the screen.
- 2) Movement time, error rate, and cursor-path efficiency are significantly affected by the task parameters such as the target width (*W*), target height (*H*), and edge position. Even though the initial cursor position is adherent to an edge, the *W* and *H* should be large as space permits.
- 3) Model variations of Fitts’ law were tested. The baseline model (Shannon formula) showed a reasonably high fit ( $R^2 = 0.959$ ), and bivariate pointing models showed higher fitness ( $R^2 = 0.966$  at most). Thus, we show that the time required for edge-to-edge pointing tasks can be estimated using current models.

## 2 RELATED WORK

### 2.1 Pointing Operations at Screen Edges

Benefits of screen edges have been investigated, such as the effects of shortening the *MT* compared to the central area of the display [3][11][12][13][14][24]. Walker and Smelcer [37][38] suggest that menus are preferred to be located at screen edges to improve usability. TorusDesktop [22] is a cursor-warping technique between screen edges, e.g., when the cursor enters the

\* syamanak@yahoo-corp.jp

right edge, it appears from the left edge to shorten the movement distance. To enable pointing to an edge target, TorusDesktop blocks the cursor for a short movement distance (125 pixels), and more pushing-edge movements allow the cursor to warp to the other-side edge.

When direct input devices (e.g., styli) are used, physical edges can stop the pen tip. This allows users to more easily draw strokes [40] and select an icon [16]. However, such benefits of edges are not observed on touchscreens using fingers. For example, Henze et al. [19] showed that touching targets on smartphone edges are prone to more errors. Avrahami [4] showed that tapping edge targets requires a longer time than targets in the central area of tablets. In summary, while touchscreens are exceptions, users can more easily point to edge targets. Hence, it is better to arrange targets that are frequently used on screen edges.

## 2.2 Performance Models on Pointing Motions

Investigating the characteristics of pointing behaviors is a core theme in HCI. Therefore, we are interested in whether edge-to-edge pointing can be explained using current models. A well-known model in HCI is the Shannon formula [26] of Fitts' law [15], which shows the  $MT$  in pointing to a target as follows:

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

where  $W$  is the target size,  $A$  is the movement distance from the initial cursor position to the target center, and  $a$  and  $b$  are empirically determined constants. The logarithmic term is called index of difficulty ( $ID$ ).

Because typical targets on GUIs are rectangular, the other dimensional size, or target height  $H$ , also significantly affects user performance. In this paper, as in the previous work by Appert et al. [3], we define  $W$  as the target size along an edge and  $H$  as that perpendicular to the edge, as shown in Figure 2.

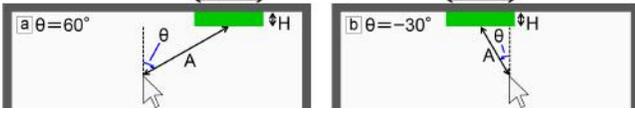


Figure 2: Definitions of  $A$ ,  $W$ ,  $H$ , and  $\theta$ .

Crossman [10] proposed the following two formulae by summing or separating the two  $ID$  values for  $W$  and  $H$  as follows:

$$MT = a + b \left[ \log_2 \left( \frac{A}{W} \right) + \log_2 \left( \frac{A}{H} \right) \right] \quad (2)$$

$$MT = a + b \log_2 \left( \frac{A}{W} \right) + c \log_2 \left( \frac{A}{H} \right) \quad (3)$$

where  $c$  is an empirically determined constant. Welford [39] proposed to separate the term of  $A/W$  in 1D tasks to capture the initial ballistic movement and final visually controlled movement:

$$MT = a + b \log_2(A) + c \log_2(W) \quad (4)$$

and this separation is theoretically supported by Meyer et al.'s [30] optimized sub-movements model. To extend this formula for 2D tasks, a potential way would be adding the term of  $\log_2(H)$  to capture the difficulty of perpendicular movements as follows:

$$MT = a + b \log_2(A) + c \log_2(W) + d \log_2(H) \quad (5)$$

where  $d$  is an empirically determined constant. MacKenzie and

Buxton [25] and Hoffmann and Sheikh [20] independently found that using the smallest value of  $W$  and  $H$  as the target size shows a good fit for bivariate (2D) pointing tasks:

$$MT = a + b \log_2 \left( \frac{A}{\min(W, H)} + 1 \right) \quad (6)$$

Note that Hoffmann and Sheikh [20] did not include the “+1” term (see the discussions in [21] and [29] for details of the additional constant of “+1”).

Accot and Zhai [1] proposed another model for 2D tasks:

$$MT = a + b \log_2 \left( \sqrt{\left( \frac{A}{W} \right)^2 + c \left( \frac{A}{H} \right)^2} + 1 \right) \quad (7)$$

where  $c$  ranged approximately from 1/3 to 1/7, which means that the perpendicular accuracy ( $H$ ) has less effect compared to the main movement accuracy ( $W$ ). Appert et al. [3] proposed the following formula for pointing to targets on screen edges:

$$MT = a + b \log_2 \left( \frac{A}{W} + \frac{A}{H} + \frac{0.6 \cdot \sin(|\theta|) \cdot A}{\min(W, H)} + 1 \right) \quad (8)$$

where  $\theta$  is defined as shown in Figure 2. In our present focus of pointing to targets on the same edge, the absolute  $\theta$  is always  $90^\circ$ . Hence, Eq. 8 for our experimental data can be converted into

$$MT = a + b \log_2 \left( \frac{A}{W} + \frac{A}{H} + \frac{0.6A}{\min(W, H)} + 1 \right) \quad (9)$$

Yang and Xu [44] found that the following simpler formula is sufficient to model 2D pointing tasks:

$$MT = a + b \log_2 \left( \frac{1}{2} \sqrt{\left( \frac{A}{W} \right)^2 + \left( \frac{A}{H} \right)^2} + 1 \right) \quad (10)$$

Finally, Zhang et al. [47] proposed to balance the effects of  $W$  and  $H$  as follows:

$$MT = a + b \log_2 \left( \sqrt{c \left( \frac{A}{W} \right)^2 + (1-c) \left( \frac{A}{H} \right)^2} + 1 \right) \quad (11)$$

where  $0 < c < 1$ . See each paper for the derivations of the models. Note that, except for Appert et al.'s [3] model (Eqs. 8 and 9), the main focus of bivariate models is estimating the  $MT$  for pointing to targets in the central area of the screen.

Appert et al.'s [3] intent was to capture edge-pointing tasks with various angles from  $\theta = -90^\circ$  to  $+90^\circ$  with steps of  $30^\circ$ . We are also interested in the effect of movement angle on user performance, but to focus on our fundamental research interest (i.e., to analyze the pushing-edge behavior as shown in Figure 1), we tested  $\theta = -90^\circ$  and  $+90^\circ$  conditions.

Appert et al. [3] also analyzed off-screen movements, i.e., the cursor's virtual movements assuming that there is no display edge. The results indicate that  $\theta = 0^\circ$  (straight approaching towards the edge) requires the longest off-screen movement distance after the cursor stops at the edge. This is consistent with Schmidt et al. [33] and Accot and Zhai's [1] studies in that the variability is larger for the main movement direction than that for the orthogonal direction. Therefore, off-screen cursor movements have been investigated by Appert et al. [3] under certain conditions, but we would like to analyze such off-screen movements in more detail. For example, to avoid clicking outside the target, users would move the cursor toward outside the screen, and thus  $H$  will significantly affect user behavior. If so, is the off-screen movement distance significantly affected by  $H$ ? We assume that the degree of the intended “safety” depends on each user's strategy, target size, and edge position. To achieve this goal, we analyze both qualitative and quantitative results.

### 3 RESEARCH QUESTIONS

#### 3.1 Off- and On-screen Cursor Movements

As described above, users can easily point to edge targets. Thus, arranging frequently selected targets on the screen edges or corners is recommended as a design guideline [18][36]. However, one drawback in edge-to-edge pointing movements would be that the pushing-edge behavior lengthens the physical movement distance of the mouse and hand. An example illustrating this drawback is shown in Figure 3. In this situation, the user moves the mouse along a natural arc-like path, and the cursor virtually follows the mouse movement (red line) if there is no display edge, but the actual cursor on the screen is blocked by the edge.

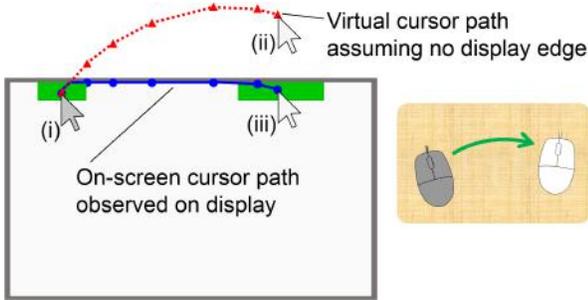


Figure 3: Cursor path comparison where the user performs an arc-like movement. Each point (triangle and circle) denotes sampled cursor trajectories. The virtual cursor movement shows an arc-like shape from (i) to (ii) following mouse movement if there is no display edge, but the cursor on the screen is blocked by the display edge and most part of the path is horizontal (from (i) to (iii)).

As shown by Appert et al. [3], we assume that  $H$  (the size on the y-axis in Figure 3) would affect user performance, because users might move the mouse more toward outside the screen to avoid missing the target as  $H$  decreases. This assumption will be discussed based on our experimental results of *path efficiency* ( $PE$ ). In addition, because the arc-like shape of human arm movements and wrist rotations are convex towards the front side (*North* of the screen), pushing-edge behavior might be more used for some edges than others. Thus, we tested all four edge positions.

#### 3.2 Path Efficiency

We discuss the  $MT$  and error rate as measures of user performance, as in previous studies. We also discuss  $PE$ . As shown in Figure 3, both the total distance including the virtual path (red; *virtual travel distance*) and that including only the on-screen path (blue; *on-screen travel distance*) are recorded. To calculate the efficiency of the cursor movements, we define  $PE$  as follows:

$$PE = \frac{\text{on-screen travel distance}}{\text{virtual travel distance}} \times 100\% \quad (12)$$

We do not use the straight-line distance from (i) to (ii) for *virtual travel distance* because the straight line ignores the effect of hand fatigue due to the total travel distance. This is the same for *on-screen travel distance*, rather than using the straight-line distance from (i) to (iii). Because one of our present interests is to determine how much extra effort (e.g., travel distance) is required to safely move the cursor onto an edge target, Eq. 12 is more appropriate than calculating the linearity of a single path proposed by [28]. For this goal, we consider that directly comparing the distances of the entire virtual and on-screen trajectories would be straightforward.

If the mouse operation shown in Figure 3 is performed in the central area of the display and the cursor is not blocked by display edges, the red and blue lines completely match. Thus, there is no wasted trajectory due to edge blocking ( $PE = 100\%$ ). On the basis of this definition, a lower  $PE$  means that a user exhibits more pushing-edge behavior, and thus s/he pays more attention to avoid the risk of clicking outside the target. A higher  $PE$  means that s/he focuses on shortening the off-screen travel distance, but it would potentially require more careful mouse operations to avoid a pointing error.

### 4 EXPERIMENT

#### 4.1 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 (model 2407WFPb; 24" 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 at 60 Hz. The input device was an optical mouse, Logitech Gaming Mouse (model G300r, 249g, 1000 Hz polling rate, 2500 dpi; 2.0-m cable). The mouse-cursor speed via the OS setting was set as the default, i.e., the control-display gain was the middle of the slider. Pointer acceleration, or the *Enhance pointer precision* setting in Windows 10, was enabled to allow the participants to perform mouse operations with higher ecological validity [7]. Using the pointer acceleration was consistent with Appert et al.'s study [3] and consumer OS settings such as Windows and Mac. We used a large mousepad, Logitech Hard Gaming Mouse Pad (model G440, 34 cm × 28 cm).

The experimental system was implemented with Hot Soup Processor 3.4 and used in full-screen mode. The system reads and processes input at approximately 500 times per second. Because the system latency would affect user performance in mouse pointing tasks [27][35], we measured the end-to-end latency by replicating Casiez et al.'s [8] method. The system moved a texture on the screen then sensed the cursor-movement event via the mouse on the display, allowing the measuring of time before *Window repaint to MouseMove event dispatched*. We repeatedly measured the latency 1000 times at nine positions on the screen ( $\{\text{leftmost, middle, and rightmost on the x-axis}\} \times \{\text{topmost, middle, and bottommost on the y-axis}\}$ ). All the 9000 trials were properly sensed, and the average latency was 56.6 ms ( $SD = 20.4$ ). This is in the range of typical mouse-display latencies of approximately 55 to 80 ms [8]. Therefore, we assume that the latency of our experimental system did not have a significant negative effect on user performance.

Appert et al. [3] compared arrow- and circle-shaped cursors (Figures 4a and b). Because an arrow cursor can hardly be seen at the *South* (bottom) and *East* (right) edges, a circular cursor is better. However, in our pilot test, some participants claimed that they could not detect where the cursor would click. To allow participants to distinguish the hotspot, we added a diagonal crosshair ("×") because the horizontal line in "+" could not be seen at *North* and *South* edges of the display.

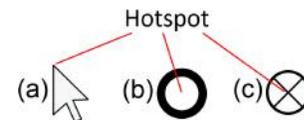


Figure 4: Cursor comparison. Appert et al. [3] tested (a) arrow and (b) thick circle shaped cursors, and (c) we used a cursor of a diagonal crosshair with a circle (25-pixel radius).

## 4.2 Participants

Twelve local university students participated in the experiment, of which four were female and eight male. The average age was 21.5 years ( $SD = 1.38$ ). All participants had normal or corrected-to-normal vision and were right-handed. Their input devices for daily PC operations (multiple choices allowed) were touchpads (12 persons), mice (four), touchscreens (two), and pen tablets (one).

## 4.3 Task

The task was to click at the start position then click inside the edge target. Because we assumed that the initial cursor position would affect the cursor trajectory and user performance, we rigorously controlled the start position. The process of one trial is shown in Figure 5. First, (a) participants clicked on the pink circle (251-pixel radius), then (b) the cursor automatically moved to the start position. Until participants pressed the mouse button again, the cursor could not be moved; the cursor always began to move from the start position controlled by the experimental system. After (c) clicking at the start position, (d) participants moved the cursor to the target and clicked on it. Data recording (time, error, and cursor trajectory) started when the start position was clicked (Figure 5c). When participants clicked correctly on the green target, a pleasant bell sound was played. Otherwise, an unpleasant beep was played to inform of an error, and then the participant had to retry the same task again (from Figure 5a).

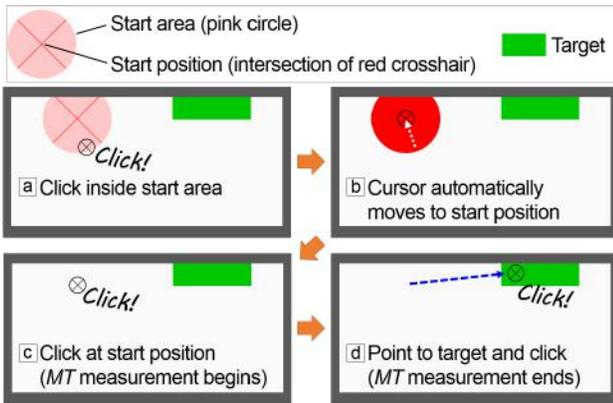


Figure 5: Experimental task where  $Edge = North$  and  $Dir = Right$ . Participants first click on the pink circle, then the cursor automatically moves to the center of the pink circle, which is the start position indicated with a red crosshair. Participants click at the start position then click on the green target.

We tested two conditions on the start position. Although Appert et al. [3] tested conditions of  $\theta = \pm 90^\circ$ , there are more potential design spaces of the initial cursor position of  $\theta = \pm 90^\circ$ . Compared to a condition in which the cursor initially contacted an edge (Figure 6a,  $Start = Contact$ ), we assume that participants tended to move the cursor more upwards under the condition in which the cursor was far from the edge (Figure 6b,  $Start = Far$ ) to avoid an error. Note that, even for  $Start = Far$ , participants just had to move the cursor horizontally; this is the most efficient trajectory in terms of movement distance. However, because such a completely horizontal movement would be difficult to perform, we assume that the  $Start$  conditions would affect participants' behavior. We were also interested in other start positions (e.g., middle of  $Contact$  and  $Far$ ), but we only tested two extreme cases to reduce the total number of task-parameter combinations.

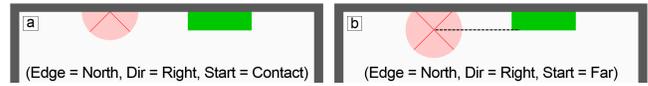


Figure 6: Comparison of start positions in the case of  $Edge = North$  and  $Dir = Right$ . (a) For  $Start = Contact$ , the start position is aligned to the  $North$  edge. (b) For  $Start = Far$ , the start position is aligned to the farthest point of the target from the edge.

## 4.4 Design and Procedure

Table 1 shows the six task parameters in this study. We tested all four  $Edge$  conditions of the rectangular screen:  $North$  (top),  $South$  (bottom),  $West$  (left), and  $East$  (right). The movement distance ( $A$ ) was measured from the start position to the target center collinearly to the movement direction. The  $W$  values from 24 to 216 pixels covered small target widths (e.g., notification icons and quick launch icons in Windows) to large ones (e.g., Chrome tabs and Windows task bars). The  $H$  values from 15 to 63 pixels covered small target heights (e.g., menu bars in Mac) to large ones (e.g., task bars arranged on the  $West$  edge in Windows 10). Two  $Start$  positions ( $Contact$  and  $Far$ ) were tested, as described in Section 4.3. Finally, for  $Edge = North$  and  $Edge = South$  conditions, we tested both main movement directions ( $Dir$ ):  $Left$  and  $Right$ . Similarly, for  $Edge = West$  and  $Edge = East$  conditions, we tested  $Dir = Up$  and  $Down$ . Note that, because comparing data under (for example)  $Dir = Left$  versus  $Dir = Up$  is meaningless,  $Dir$  is not included as a dependent variable in ANOVA. Figures 7 show two example tasks. Again, the notation  $W$  denotes the target size along the edge, and  $H$  denotes the orthogonal size to the edge.

One block consisted of a random ordering of  $4(Edge) \times 2(A) \times 3(W) \times 3(H) \times 2(Start) \times 2(Dir) = 288$  conditions. Participants performed 20 trials randomly selected from these conditions for training, then three blocks for data collection. The recorded data were  $288 \text{ conditions} \times 3 \text{ blocks} \times 12 \text{ participants} = 10368$  trials. After completing three blocks, we interviewed the participants on their impressions and strategies. Each participant took approximately 50 minutes for this study.

Participants were instructed to select the target as quickly and accurately as possible after clicking at the start position. In addition, to avoid clutching actions (i.e., repositioning the mouse on the mousepad) during data measurement, they were encouraged to reposition the mouse before they clicked at the start position. They were also permitted to adjust the chair height, display angle, and mousepad position for comfortableness.

Table 1: Task parameters used in the experiment.

| Edge    | <i>North, South, West, and East</i>   |
|---------|---|
| $A$     | 300 and 700 pixels  |
| $W$     | 24, 72, and 216 pixels  |
| $H$     | 15, 31, and 63 pixels   |
| $Start$ | <i>Contact and Far</i>  |
| $Dir$   | <i>Left and Right (for Edge = North or South)<br/>Up and Down (for Edge = West or East)</i> |

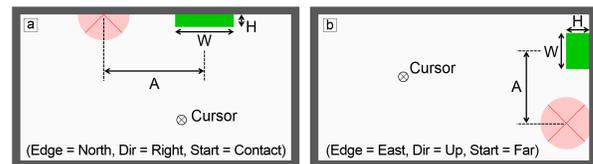


Figure 7: Examples of task-parameter combinations.

## 5 RESULTS

We recorded 10749 trials including 381 error instances (3.54%) due to clicking outside the target. This error rate was close to that in typical pointing tasks of 4% [26][41][45][46]. Therefore, our experimental settings (target size, mouse cursor speed, and so on) were appropriately customized for observing the characteristics of pointing operations.

Although some researchers [10][26][34][41] and the ISO standard 9241-9: 2000 [23] recommend using the *effective width method* (i.e., correcting the data of target size based on the spread of clicking positions) to calculate throughputs, we used only nominal parameter values for comparing Fitts' law fitness among candidate models. We analyzed only error-free trials by *mean-of-means* calculation via repeated-measures ANOVA with the Bonferroni post-hoc test. Again, we did not include *Dir* as a dependent variable.

### 5.1 Movement Time (MT)

Figure 8 shows the results of *MT*. We observed the main effects of *Edge* ( $F_{3,33} = 9.919, p < 0.001$ ), *A* ( $F_{1,11} = 650.471, p < 0.001$ ), *W* ( $F_{2,22} = 509.707, p < 0.001$ ), *H* ( $F_{2,22} = 39.655, p < 0.001$ ), and *Start* ( $F_{1,11} = 8.066, p < 0.05$ ). Pair-wise comparisons indicate that the *MT* for *Edge = North* was the smallest ( $p < 0.01$  for the other three edges), and the other three edges showed no significant difference ( $p > 0.05$  for each). These results also indicate that *MT* increased as *A* increased ( $p < 0.001$ ), *W* decreased ( $p < 0.001$  for all pairs), and *H* decreased ( $p < 0.001$  for all pairs). These results on *A*, *W*, and *H* are consistent with those from previous studies, which means that *MT* increases as the task difficulty increases.

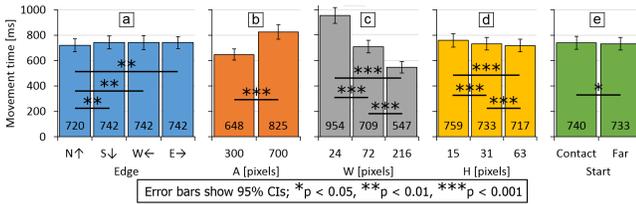


Figure 8: Results of *MT* for each parameter. N: North, S: South, W: West, and E: East in (a).

### 5.2 Error Rate

We observed the main effect of *W* ( $F_{2,22} = 7.730, p < 0.01$ ). Pair-wise comparisons indicate that error rates increased as *W* decreased ( $p < 0.01$  for all pairs). The other task parameters did not show the main effects: *Edge* ( $F_{3,33} = 2.017, p = 0.131$ ), *A* ( $F_{1,11} = 0.672, p = 0.430$ ), *H* ( $F_{2,22} = 0.508, p = 0.608$ ), and *Start* ( $F_{1,11} = 2.554, p = 0.138$ ). We observed a significant interaction of  $H \times Start$  ( $F_{2,22} = 3.697, p < 0.05$ ), but the other parameter combinations were not significant ( $p > 0.05$ ).

### 5.3 Path Efficiency (PE)

Figure 9 shows the results of *PE*. We observed the main effects of *Edge* ( $F_{3,33} = 13.976, p < 0.001$ ), *A* ( $F_{1,11} = 48.708, p < 0.001$ ), *W* ( $F_{2,22} = 39.148, p < 0.001$ ), *H* ( $F_{2,22} = 248.623, p < 0.001$ ), and *Start* ( $F_{1,11} = 18.427, p < 0.01$ ). The *PE* increased as *A* increased ( $p < 0.001$ ), *W* decreased ( $p < 0.001$  to 0.01), and *H* increased ( $p < 0.001$  for all pairs). We observed no significant interaction. Among all participants, *PE* ranged from 95.3 to 98.7% (participants #11 and #5), which means that participants could not completely avoid pushing-edge behavior. Throughout the experiment, we observed 6,022,474 pixels of *on-screen travel distance*, and 6,210,568 pixels of *virtual travel distance*, resulting

in  $PE = 97.0\%$  on average due to the 188,094-pixel difference.

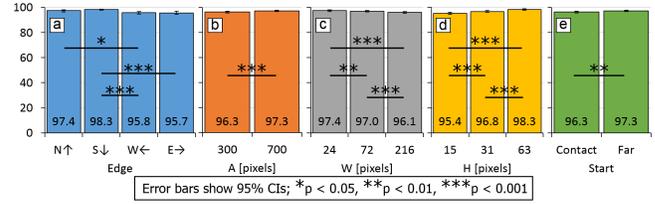


Figure 9: Results of *PE* for each parameter.

## 6 DISCUSSION

### 6.1 User Strategy and Cursor Trajectory

In accordance with the oral interview after the experiment, 6 out of 12 participants stated that they kept on moving the cursor and contacting the edge. However, task parameters affected their pushing-edge behavior. One (#6) of these six participants stated that he exhibited this behavior for vertical directions (*Dir = Up* and *Down*), but he could not smoothly exhibit it for horizontal movements (*Dir = Left* and *Right*). A reason behind this difference would be that comfortable arm and hand movements depend on the direction; horizontal directions are more affected by arc-like hand movements, but vertical directions would require more intentional control by the user.

Similarly, another participant (#12) among the six participants stated that downward movements at the right edge (*Dir = Down* and *Edge = East*) were the easiest for exhibiting pushing-edge behavior. However, when the condition was horizontal movements (*Dir = Left* and *Right*), he first moved the cursor towards the central area then pointed to the target, because he thought that it was the most natural movement for him. This behavior can be seen in his cursor trajectory shown in Figure 10a; although the start position contacted the *North* edge, he first left the edge then moved toward the target. He also sometimes did not contact the edge when *Start = Far* (Figure 10b). However, he also exhibited typical pushing-edge behavior for *Edge = North* conditions, as shown in Figures 10c and d. Hence, we could not determine a consistent tendency in pushing-edge behavior.

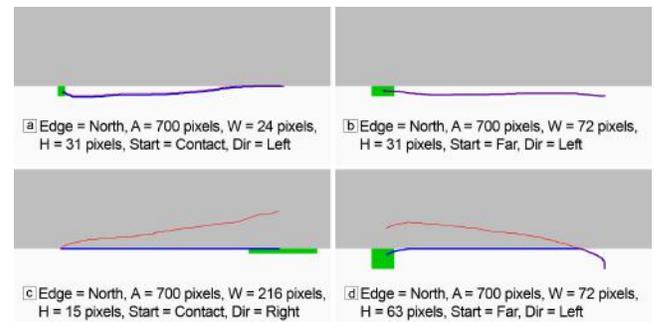


Figure 10: Example cursor trajectories (from the start position to green target) made by participant #12. Red lines show the virtual cursor trajectories assuming there was no screen edge, and blue lines show the on-screen trajectories blocked by the edge. Gray areas show the off-screen area.

Among the other six participants who did not state that they exhibited pushing-edge behavior, one (#3) stated that he intentionally avoided contacting the edge, similarly to Figure 10b. The reason was that “[For *Edge = North*,] if I moved the cursor diagonally upwards a little, it would require a longer distance and take a longer time.” However, he also stated that “when the target

size [ $W$  or  $H$ ] was small, it was okay to contact the edge because ‘additional movements to avoid contacting the edge’ in such conditions would take an even longer time.”

In accordance with the participants’ statements above and trajectory analysis, our assumption that users would change their behavior depending on the task requirements is supported. Among all 288 task conditions, the highest and lowest  $PE$ s were 99.8% for ( $Edge = North, A = 300$  pixels,  $W = 216$  pixels,  $H = 15$  pixels,  $Start = Contact, Dir = Right$ ) and 89.0% for ( $Edge = South, A = 700$  pixels,  $W = 24$  pixels,  $H = 63$  pixels,  $Start = Far, Dir = Right$ ). Two participants (#3 and #12) stated the advantage of avoiding contacting the edge, but  $PE$ s for the two participants were 96.7 and 97.1%, respectively. Hence, even if users would not like to waste their movements off the screen, approximately 3–4% of cursor travel distance was not reflected on the screen in edge-to-edge pointing tasks under our experimental conditions. To achieve more efficient mouse movements and shorter operation times, the  $W$  and  $H$  should be large as space permits, even if the initial cursor position is adherent to an edge.

Another interesting comment was that the pushing-edge behavior allowed the manually changing of the control-display gain. In other words, as the user moves the cursor more toward outside the screen, the on-screen cursor movement distance decreases. Because it was shown to be better to increase the gain for initial ballistic movements and decrease the gain for final cursor adjustments around the target [5][42], pushing-edge behavior would be beneficial for dynamic gain tuning. Unfortunately, we could not re-analyze how the participants dynamically changed the gain in a single trial, because we enabled the pointer acceleration. We assume that conducting an experiment with constant gain would provide additional contributions for better understanding edge-to-edge pointing tasks by analyzing the dynamic gain changes.

## 6.2 Model Fitting

Table 2 lists the results of model fitting using all (288) data points. In addition to comparing adjusted  $R^2$  data, we show Akaike information criterion ( $AIC$ ) values [2][6].  $AIC$  balances the complexity of the model (the number of free parameters) and the fitness, and determines the comparatively best model. A model with (a) lower  $AIC$  value is a better one, (b)  $AIC$  value  $\leq$  (minimum- $AIC$  value + 2) considers comparisons with better models, and (c)  $AIC$  value  $\geq$  (minimum- $AIC$  value + 10) is safely rejected. This analysis method has been used to evaluate performance models for GUI operations [9][32][43][48]. Based on the obtained  $R^2$  values, Accot and Zhai’s model (Eq. 7) and Zhang et al.’s model (Eq. 11) can most accurately capture user performance. This may be simply because they have more degrees of freedom than the baseline model (Eq. 1). Still, they also show the lowest  $AIC$  values, and thus they are statistically the best models among the ten candidates.

A difference from Appert et al.’s [3] analysis is that we used nominal  $H$  values, while Appert et al. used the approximation for  $H$ . In Appert et al.’s study,  $H$  was set to 320 pixels on screen, and more than 99% of click positions were observed in the first 250 pixels. Therefore, Appert et al. used  $H = 250$  pixels for model fitting. This could be seen as a kind of *posteriori* data correction similarly to the *effective width method*, but we compared the data using nominal  $A$ ,  $W$ , and  $H$ . For this reasoning, we do not include  $PE$  as a predictor, e.g.,  $MT = a + b \log_2(A/W + 1) + cPE$ . In fact, this model shows a good fit: adjusted  $R^2 = 0.964$  and  $AIC = 2904$ .

It should be noted that the baseline formula (Eq. 1) is already a reasonable model for pointing actions between targets on the same edge ( $R^2 = 0.959$ ). A reason behind this would be that we only

tested conditions of  $\theta = \pm 90^\circ$ . These conditions resemble a traditional 1D Fitts’ pointing task that requires adjusting the cursor position collinearly to the main movement direction more than the orthogonal direction. This is also supported by the  $c$  value of Eq. 11; weights for  $(A/W)^2$  versus  $(A/H)^2$  are 0.994:0.006. On the contrary, Appert et al.’s model for edge-pointing actions (Eq. 9) does not show a good fit ( $R^2 = 0.481$ ). We assume the reason would be that Eq. 9 (and the original formula of Eq. 8) was derived to capture user performance under various approaching angles ranging from  $\theta = -90^\circ$  to  $+90^\circ$ . In summary, for edge-to-edge pointing tasks, using only one target-size parameter along the edge ( $W$ ) have a sufficient level to estimate  $MT$ , and adding the other dimensional size ( $H$ ) can statistically improve fitness.

## 6.3 Limitations and Future Work

Our results are somewhat limited by the experimental conditions, such as we tested only two directions ( $\theta = -90^\circ$  and  $+90^\circ$ ). In addition, we did not test a special case of edge targets: targets on corners. Because a target on a screen corner can be pointed to without precise mouse control,  $W$  and  $H$  would have less effect on  $MT$  [17]. Thus, the  $MT$  for such actions would depend only on  $A$ :

$$MT = a + b\sqrt{A} \quad (13)$$

but more empirical evidence is necessary to confirm this model for corner targets.

Another limitation is that we only tested a mouse. Other devices, such as trackballs and touchpads, would reveal different characteristics from mice, in particular for cursor trajectories. A typical difference will come from clutching motions; we encouraged the participants to avoid repositioning the mouse during a trial, but common touchpads require one or more clutches for a single target-pointing trial [31]. Hence, we cannot assume that the results obtained in our study can be applied to other input devices.

We would like to investigate other factors lowering  $PE$ . For example, for a short  $A$  (e.g., 30 pixels), we assume that users do not have to exhibit pushing-edge behavior. In comparison, for a long  $A$  (e.g., 1500 pixels), the  $PE$  would be low because users would like to prevent the cursor from moving away from the edge. In addition, when users horizontally move the cursor, they tend to pivot their wrists. This induces arc-like mouse movements rather than straight lines, and the  $PE$  would be lower as  $A$  increases at a screen edge. However, Figure 9b, which shows that a longer  $A$  achieves a higher  $PE$ , rejects this assumption. We want to uncover the cause of this by conducting an experiment including extremely short and long  $A$  values. Note that arc-like movements do not necessarily mean that effort is being “wasted,” because this way of moving may be more natural for users. Hence, we also want to study the relationship between  $PE$  and users’ subjective comfortableness when they perform pushing-edge behavior.

We are also interested in testing performance models on physical edge-to-edge pointing tasks such as Barrier Pointing [16]. With this technique, users can move a stylus tip (or probably a finger tip) on the surface along a physical edge then select a button by releasing the stylus tip. If the initial position of the stylus tip is above a previously selected edge button, this task can be similar to that in our experiment. While we cannot measure the virtual travel distance in such physical-edge pointing tasks, we can measure different aspects of user behavior, e.g., when and why users would apply more pressure to the physical edge, which can be measured with an additional pressure sensor. Empirical data will provide interesting user behavior on edge-to-edge pointing operations on pen tablets and touchscreens.

Table 2: Adjusted  $R^2$  and  $AIC$  values for candidate models. Estimated  $a$ ,  $b$ ,  $c$ , and  $d$  are constants with 95% CIs [lower, upper]. Colored cells show best values for each analysis method (adjusted  $R^2$  and  $AIC$ ).

| Model   | Eq. | $a$                   | $b$                  | $c$                           | $d$                     | $R^2$ | $AIC$ |
|---|-----|-----------------------|----------------------|-------------------------------|-------------------------|-------|-------|
| $MT = a + b \log_2 \left( \frac{A}{W} + 1 \right)$  | 1   | 272<br>[256, 284]     | 157<br>[153, 160]    |                               |                         | 0.959 | 2935  |
| $MT = a + b \left[ \log_2 \left( \frac{A}{W} \right) + \log_2 \left( \frac{A}{H} \right) \right]$                     | 2   | 166<br>[130, 201]     | 86.9<br>[81.7, 92.1] |                               |                         | 0.792 | 3402  |
| $MT = a + b \log_2 \left( \frac{A}{W} \right) + c \log_2 \left( \frac{A}{H} \right)$                                  | 3   | 321<br>[304, 338]     | 128<br>[125, 131]    | 18.9<br>[14.8, 23.1]          |                         | 0.965 | 2893  |
| $MT = a + b \log_2(A) + c \log_2(W)$  | 4   | 248<br>[245, 252]     | 145<br>[73.8, 216]   | -128<br>[-136, -121]          |                         | 0.957 | 2950  |
| $MT = a + b \log_2(A) + c \log_2(W) + d \log_2(H)$  | 5   | 347<br>[279, 416]     | 145<br>[138, 152]    | -128<br>[-132, -125]          | -20.0<br>[-25.0, -15.0] | 0.965 | 2895  |
| $MT = a + b \log_2 \left( \frac{A}{\min(W, H)} + 1 \right)$   | 6   | 232<br>[145, 319]     | 120<br>[100, 141]    |                               |                         | 0.326 | 3741  |
| $MT = a + b \log_2 \left( \sqrt{\left( \frac{A}{W} \right)^2 + c \left( \frac{A}{H} \right)^2} + 1 \right)$           | 7   | 235<br>[220, 250]     | 165<br>[161, 169]    | 0.00553<br>[0.00337, 0.00730] |                         | 0.966 | 2882  |
| $MT = a + b \log_2 \left( \frac{A}{W} + \frac{A}{H} + \frac{0.6A}{\min(W, H)} + 1 \right)$                            | 9   | -12.5<br>[-105, 79.7] | 146<br>[128, 163]    |                               |                         | 0.481 | 3666  |
| $MT = a + b \log_2 \left( \frac{1}{2} \sqrt{\left( \frac{A}{W} \right)^2 + \left( \frac{A}{H} \right)^2} + 1 \right)$ | 10  | 232<br>[162, 302]     | 148<br>[128, 168]    |                               |                         | 0.424 | 3696  |
| $MT = a + b \log_2 \left( \sqrt{c \left( \frac{A}{W} \right)^2 + (1-c) \left( \frac{A}{H} \right)^2} + 1 \right)$     | 11  | 235<br>[221, 250]     | 165<br>[161, 169]    | 0.994<br>[0.993, 0.996]       |                         | 0.966 | 2882  |

## 7 CONCLUSION

We empirically studied the characteristics of pointing operations for targets on the same screen edge. Participants tended to move the mouse toward outside the screen edge to accurately click on the target, and this pushing-edge behavior induced a long *virtual travel distance* of the cursor. Overall, the *PE* (i.e., the ratio of *on-screen travel distance* divided by *virtual travel distance*) was approximately 97% on average, but the *PE* had a certain range from 89.0 to 99.8% depending on the task conditions. We also tested the adequateness of current performance models on Fitts' law. The baseline model (Shannon formula, Eq. 1) could accurately capture the edge-to-edge pointing motions, and its variations considering bivariate parameters (Eqs. 7 and 11) were the best models regarding the higher  $R^2$  and lower  $AIC$  values.

Quantitative results also supported that some participants changed their behavior depending on the task requirements such as the edge position (*North* versus *West*, etc.). In accordance with our results, as in Appert et al.'s study [3], the target size on the screen (both  $W$  and  $H$ ) should be large as space permits, which would achieve a shorter  $MT$  (Section 5.1) and higher  $PE$  (Section 5.3) without significant increase in error rate (Section 5.2). Testing targets on corners and other devices will provide much better understanding of user performance to improve GUIs in traditional desktop environments.

## REFERENCES

- [1] Johnny Accot and Shumin Zhai. 2003. Refining Fitts' law models for bivariate pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '03), 193–200.
- [2] Hirotugu Akaike. 1974. A new look at the statistical model identification. *IEEE Transaction on Automatic Control*, Vol.19, No.6, 716–723.
- [3] Caroline Appert, Olivier Chapuis, and Michel Beaudouin-Lafon. 2008. Evaluation of pointing performance on screen edges. In *Proceedings of the working conference on Advanced visual interfaces* (AVI '08), 119–126.
- [4] Daniel Avrahami. 2015. The effect of edge targets on touch performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '15), 1837–1846.
- [5] Renaud Blanch, Yves Guiard, and Michel Beaudouin-Lafon. Semantic pointing: improving target acquisition with control-display ratio adaptation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '04), 519–526.
- [6] Kenneth P. Burnham and David R. Anderson. 1998. *Model selection and multimodel inference: a practical information-theoretic approach*. Springer.
- [7] Géry Casiez and Nicolas Roussel. 2011. No more bricolage!: methods and tools to characterize, replicate and compare pointing transfer functions. In *Proceedings of the annual ACM symposium on User interface software and technology* (UIST '11), 603–614.
- [8] Géry Casiez, Stéphane Conversy, Matthieu Falce, Stéphane Huot, and Nicolas Roussel. 2015. Looking through the eye of the mouse: a simple method for measuring end-to-end latency using an optical mouse. In *Proceedings of the annual ACM symposium on User interface software and technology* (UIST '15), 629–636.
- [9] Olivier Chapuis and Pierre Dragicevic. 2011. Effects of motor scale, visual scale, and quantization on small target acquisition difficulty. *ACM Transactions on Computer-Human Interaction* (TOCHI), Vol.18, No.3, Article 13.
- [10] E.R.F.W. Crossman. 1956. The measurement of perceptual load in manual operations. Unpublished Ph.D. thesis. University of Birmingham.
- [11] J. Shawn Farris, Keith S. Jones, Brent A. Anders. 2001. Acquisition speed with targets on the edge of the screen: an application of Fitts' law to commonly used web browser controls. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (HFES '01), 1205–1209.
- [12] J. Shawn Farris, Keith S. Jones, Brent A. Anders. 2002a. Using

- impenetrable borders in a graphical web browser: are all angles equal? In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (HFES '02), 1251–1255.
- [13] J. Shawn Farris, Keith S. Jones, Brent A. Anders. 2002b. Using impenetrable borders in a graphical web browser: how does distance influence target selection speed? In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (HFES '02), 1300–1304.
- [14] J. Shawn Farris, Keith S. Jones, Brent A. Anders. 2002c. Factors affecting the usefulness of impenetrable interface element borders. *Human Factors*, Vol.44, No.4, 578–591.
- [15] Paul M. Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, Vol.47, No.6, 381–391.
- [16] Jon Froehlich, Jacob O. Wobbrock, and Shaun K. Kane. 2007. Barrier pointing: using physical edges to assist target acquisition on mobile device touch screens. In *Proceedings of the international ACM SIGACCESS conference on Computers and accessibility* (ASSETS '07), 19–26.
- [17] Khai-Chung Gan and Errol R. Hoffmann. 1988. Geometrical conditions for ballistic and visually controlled movements. *Ergonomics*, Vol.31, No.5, 829–839.
- [18] Jensen Harris. 2006. Giving You Fitts. An Office User Interface Blog, <https://blogs.msdn.microsoft.com/jensenh/2006/08/22/giving-you-fitts/> (Retrieved December 20, 2017)
- [19] Niels Henze, Enrico Rukzio, and Susanne Boll. 2011. 100,000,000 taps: analysis and improvement of touch performance in the large. In *Proceedings of the International Conference on Human Computer Interaction with Mobile Devices and Services* (MobileHCI '11), 133–142.
- [20] Errol R. Hoffmann and Ilyas H. Sheikh. 1994. Effect of varying target height in a Fitts' movement task. *Ergonomics*, Vol.37, No.6, 1071–1088.
- [21] Errol R. Hoffmann. 2012. Which version/variation of Fitts' law? A critique of information-theory models. *Journal of Motor Behavior*, Vol.45, No.3, 205–215.
- [22] Stéphane Huot, Olivier Chapuis, Pierre Dragicevic. 2011. TorusDesktop: Pointing via the Backdoor is Sometimes Shorter. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '11), 829–838.
- [23] ISO 9241-9: 2000. 2000. Ergonomic requirements for office work with visual display terminals (VDTs) - Part 9: Requirements for non-keyboard input devices, International Organization for Standardization.
- [24] Brian R. Johnson, J. Shawn Farris, Keith S. Jones. 2003. Selection of web browser controls with and without impenetrable borders: does width make a difference? In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (HFES '03), 1380–1384.
- [25] I. Scott MacKenzie and William Buxton. 1992a. Extending Fitts' law to two-dimensional tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '92), 219–226.
- [26] I. Scott MacKenzie. 1992b. Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 91–139.
- [27] I. Scott MacKenzie and Colin Ware. 1993. Lag as a determinant of human performance in interactive systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '93), 488–493.
- [28] I. Scott MacKenzie, Tatu Kauppinen, and Miika Silfverberg. 2001. Accuracy measures for evaluating computer pointing devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '01), 9–16.
- [29] I. Scott MacKenzie. 2013. A note on the validity of the Shannon formulation for Fitts' index of difficulty. *Open Journal of Applied Sciences*, Vol.3, No.6, 360–368.
- [30] David E. Meyer, Richard A. Abrams, Sylvan Kornblum, Charles E. Wright, and J. E. Keith Smith. 1988. Optimality in human motor performance: ideal control of rapid aimed movements. *Psychological Review*, Vol.95, No.3, 340–370.
- [31] Mathieu Nancel, Daniel Vogel, and Edward Lank. 2015. Clutching is not (necessarily) the enemy. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '15), 4199–4202.
- [32] Xiangshi Ren, Jing Kong, and Xing-Qi Jiang. 2005. SH-model: a model based on both system and human effects for pointing task evaluation. *IPSJ Journal*, Vol.46, No.5, 1343–1353.
- [33] Richard A. Schmidt, Howard Zelaznik, Brian Hawkins, James S. Frank, and John T. Quinn, Jr. 1979. Motor-output variability: A theory for the accuracy of rapid motor acts. *Psychological Review*, Vol.86, No.5, 415–451.
- [34] R. William Soukoreff and I. Scott MacKenzie. 2004. Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI. *International Journal of Human-Computer Studies*, Vol.61, No.6, 751–789.
- [35] Robert J. Teather, Andriy Pavlovych, Wolfgang Stuerzlinger, and I. Scott MacKenzie. 2009. Effects of tracking technology, latency, and spatial jitter on object movement. In *Proceedings of the IEEE Symposium on 3D User Interfaces* (3DUI '09), 43–50.
- [36] Bruce Tognazzini. 1999. A quiz designed to give you Fitts. AskTog, <http://www.asktog.com/columns/022DesignedToGiveFitts.html> (Retrieved December 20, 2017)
- [37] Neff Walker and John B. Smelcer. 1990. A comparison of selection time from walking and pull-down menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '90), 221–226.
- [38] Neff Walker and John B. Smelcer. 1991. Optimizing speed and accuracy of menu selection: a comparison of walking and pull-down menus. *International Journal of Man-Machine Studies*, Vol.35, No.6, 871–890.
- [39] Alan Triviss Welford. 1968. *Fundamentals of Skill*. Methuen Publishing
- [40] Jacob O. Wobbrock, Brad A. Myers, and John A. Kемbel. 2003. EdgeWrite: a stylus-based text entry method designed for high accuracy and stability of motion. In *Proceedings of the annual ACM symposium on User interface software and technology* (UIST '03), 61–70.
- [41] 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* (CHI '11), 1639–1648.
- [42] Aileen Worden, Neff Walker, Krishna Bharat, and Scott Hudson. 1997. Making computers easier for older adults to use: area cursors and sticky icons. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '97), 266–271.
- [43] Shota Yamanaka, Wolfgang Stuerzlinger, and Homei Miyashita. 2017. Steering through sequential linear path segments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '17), 232–243.
- [44] Huahai Yang and Xianggang Xu. 2010. Bias towards regular configuration in 2D pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10), 1391–1400.
- [45] Shumin Zhai. 2004a. Characterizing computer input with Fitts' law parameters: the information and non-information aspects of pointing. *International Journal of Human-Computer Studies*, 791–809.
- [46] Shumin Zhai, Jing Kong, and Xiangshi Ren. 2004b. Speed-accuracy tradeoff in Fitts' law tasks: on the equivalency of actual and nominal pointing precision. *International Journal of Human-Computer Studies*, 823–856.
- [47] Xinyong Zhang, Hongbin Zha, and Wenxin Feng. 2012. Extending Fitts' law to account for the effects of movement direction on 2d pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '12), 3185–3194.
- [48] Jian Zhao, R. William Soukoreff, Xiangshi Ren, and Ravin Balakrishnan. 2014. A model of scrolling on touch-sensitive displays. *International Journal of Human-Computer Studies*, Vol.72, No.12, 805–821.