

Effect of a Cursor Warping Left and Right of the Notch

Yosuke Oba
Meiji University
Nakano-ku, Tokyo, Japan
yosuke@obaonline.net

Homei Miyashita
Meiji University
Nakano-ku, Tokyo, Japan
homei@homei.com

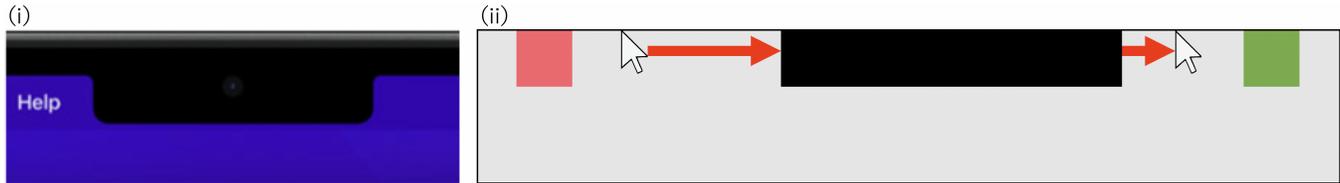


Figure 1: (i) The notch on MacBook Pro (2021). The area of the notch is not a screen and is not drawn. Although the cursor can be entered, in the case of the default size cursor, the notch hides the cursor partially or entirely. As a result, a user can lose the cursor. (ii) A cursor warping to the left and right of the notch. For example, if the cursor enters the left edge of the notch, it will warp to the right side of the notch. This cursor may shorten the path to the target and restrain the movement time increased by the notch.

ABSTRACT

Although the mouse cursor can enter the notch of the MacBook Pro (2021), it is partially or entirely hidden by the notch. Moving the cursor around the notch or avoiding it entirely can increase the movement time. This study investigated the possibility of using a cursor warping to the left and right of the notch to shorten the path to the target and restrain the movement time increased by the notch. We conducted an experiment to compare the performance of a default cursor, a cursor twice the default size, and a cursor warping to the left and right of the notch. The results showed that a cursor warping left and right of the notch was not effective in terms of the movement time.

CCS CONCEPTS

• Human-centered computing → Empirical studies in HCI.

KEYWORDS

Mouse pointing, human motor performance, notch, cursor warping, edge target, Fitts' law

ACM Reference Format:

Yosuke Oba and Homei Miyashita. 2023. Effect of a Cursor Warping Left and Right of the Notch. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems (CHI EA '23)*, April 23–28, 2023, Hamburg, Germany. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3544549.3585766>

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CHI EA '23, April 23–28, 2023, Hamburg, Germany

© 2023 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-9422-2/23/04.

<https://doi.org/10.1145/3544549.3585766>

1 INTRODUCTION

A notch is placed to expand the area of a PC screen while positioning the webcam at the same time (Fig. 2). The MacBook Pro (2021) (Apple Inc., US)¹ display has a notch at the center of the top edge as a black, non-drawing area. The cursor can enter the area of the notch in current notch specification (Fig. 1(i)). When the cursor enters the notch, it is generally partially or entirely hidden by the notch. However, there is an exception to the MacBook Pro's cursor behavior with respect to the notch: if the user has already clicked an item on the menu bar, the cursor warps to the left and right of the notch. In other words, there are currently two types of behaviors for the notch: entering the notch and warping to either side of the notch.

Pointing (point targets such as buttons or icons) should be completed quickly and accurately. The target size and distance to the target affect the movement time of the cursor operation [9, 16]. Furthermore, a smaller target and longer distance requires longer movement time. In addition, the movement time increases when distractors, which are targets that a user does not intend to select, are on the path to the target [5, 14, 22, 23]. Oba et al. [20] showed that the notch hides the cursor, which also increases the movement time. Furthermore, when the notch is placed, it is desirable for the user to avoid the notch [19]. Therefore, it is desirable to make the notch an area where the cursor cannot enter [19].

This study investigated the possibility of using a cursor warping to the left and right of the notch (Fig. 1 (ii)) to shorten the path to the target and restrain the movement time, which increases owing to the notch. In other words, it may be desirable for the cursor to always warp to the left or right of the notch, regardless of whether the user has already selected an item on the menu bar. We conducted an experiment to compare a cursor of the default size, a cursor twice the default size, and a cursor warping to the left and right of the notch. We assumed that a cursor twice the

¹Apple Inc., US. <https://www.apple.com/>



Figure 2: Overall view of the MacBook Pro display. The notch is placed at the top center of the display to extend the screen size and to accommodate the webcam.

default size might also restrain the increasing movement time, considering the entire cursor would not be hidden by the notch. The results of the experiment did not demonstrate the effectiveness of the cursor warping to the left and right of the notch. Although warping the cursor shortened the distance the user had to move the mouse and improved the efficiency of the cursor path, it did not restrain the movement time increased by the notch. This can be attributed to participants' perception of the target width, which is smaller than the actual target width, as they avoid clicking near the notch to avoid unnecessary warping. Participants reported that they carefully moved the cursor before warping because the warp caused the cursor to move abruptly. When using a warping cursor, the maximum average speed of the cursor tends to decrease. In addition, even in the no notch condition, the movement time was longer when using a warping cursor. We believe that it would be undesirable to have a mixture of warping and non-warping situations, which would require the user to vary the distance traveled by the mouse depending on the situation.

2 RELATED WORK

2.1 Effect of the Notch

Oba et al. [20] showed that the notch increases the movement time when pointing to an edge target from another edge target on the top edge of the screen. The movement time was increased primarily when the notch was between the targets. Furthermore, when the notch was between the targets, participants moved the cursor using two main strategies: (i) move the cursor along the edge (along-strategy) and (ii) avoid the notch (avoid-strategy). In the along-strategy, the cursor was hidden by the notch, and hence, the participants could not locate the cursor. In the avoid-strategy, avoiding the notch increased the path and movement time.

Furthermore, Oba et al. [19] showed that the avoid-strategy is preferred over the current notch specifications (the cursor can enter). Although the distance that participants moved the mouse was longer in avoid-strategy than in along-strategy, the movement

time was shorter for the avoid-strategy because the cursor was not hidden by the notch and the cursor was always visible. In addition, when the notch was placed on a display, it was desirable to design the notch in an area where the cursor could not enter [19]. The outer edge of the notch stops the cursor, resulting in faster and more accurate target-pointing, if the target is adjacent to the notch.

2.2 Effect of Warping the Cursor

Patrick et al. [4] observed that when multiple displays with different resolutions are used, the cursor unnaturally warps between the displays. To solve this problem, they proposed the Mouse Ether method, which eliminates unnatural warping when the cursor is moved between displays. This technique prevented unnatural warps and improved the performance by up to 28%.

Stephane et al. [13] investigated cursor warping that occurs in display torus settings (e.g., when the cursor reaches the right edge of the screen, it appears from the left edge of the screen). Therefore, they proposed the TorusDesktop method, which presents appropriate visual feedback before the warp occurs and the cursor emerges after the warp.

2.3 Effects of Edge Target and Path Efficiency

A target placed against the edge of the screen is commonly referred to as an edge target, which can be pointed faster than a target that does not contact the edge of the screen [2, 7, 8, 24]. To point at a target at the center of the screen, the user must stop the cursor within the target area. However, when pointing at an edge target, the user can take advantage of the fact that the cursor stops at the edge of the screen. As a result, edge targets can be pointed at by simply adjusting the cursor horizontally only with the edge to which the target is adjacent.

However, pushing-edge (pushing the cursor to the edge of the screen) increases the distance traveled by the mouse and decreases the efficiency of the cursor path. Yamanaka defined *PE* (Path Efficiency), which calculates the efficiency of the path of the cursor's movement [24]:

$$PE = \frac{\text{on-screen travel distance}}{\text{virtual travel distance}} \times 100\% \quad (1)$$

PE is calculated by the on-screen travel distance (the distance the cursor moves on-screen) and the virtual travel distance (the distance the virtual cursor moves while ignoring the fact that the cursor stops at the edge of the screen). If the user performs extra pushing-edge, *PE* exhibits a low value, thereby indicating that the path is less efficient. Conversely, if the user rarely performs pushing-edge, *PE* is approximately 100%, which indicates that the path is highly efficient. However, a high *PE* value does not necessarily indicate less movement time, and the user may not be able to effectively use the benefits of the edge of the screen. In this study, when a cursor warping to the left and right of the notch is used, the width of the notch is added to the on-screen travel distance when the cursor warps, resulting in a situation where the on-screen travel distance is larger than the virtual travel distance. Note that *PE* may show values greater than 100%.

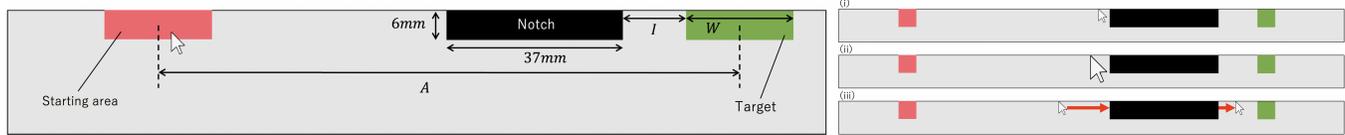


Figure 3: (Left) A schematic of the experiment. (Right) A schematic of Cursor of (i) default, (ii) twice, and (iii) warp.

2.4 Performance Models for Pointing

Fitts' law [9, 16] is a typical performance model that predicts the pointing movement time (MT). There are several formulas called Fitts' law; however, the Shannon formulation is commonly used in HCI research:

$$MT = a + bID, ID = \log_2 \left(\frac{A}{W} + 1 \right) \quad (2)$$

where a and b are empirical constants. ID (index of difficulty) indicates the difficulty level of the pointing task and is calculated from the distance from the initial position of the cursor to the target center (A) and the target width (W).

Jax et al. [14] and Vaughan et al. [23] investigated the case of distractor on the path to the target and proposed a model for when the user maneuvers around the distractor. Yamanaka et al. [25] showed that when a distractor was not on the path but near the path, the user bent the path to avoid the distractor.

Furthermore, the target height (H) also affects the movement time [2, 6, 12, 17, 18, 26]. Accot and Zhai [1] and Zhang et al. [27] proposed a model for a bivariate (2D) pointing task that takes H . Yamanaka [24] showed that the movement time for pointing to an edge target from another edge target on the same edge can be predicted using 2D models. Usuba et al. [22] proposed a model that considers the interval between the target and the distractor (I). By combining the models proposed by Zhang et al. [27] and Usuba et al. [22], Oba et al. [20] proposed a model that considers the placement of the notch when pointing to an edge target from another edge target on the top edge of the screen.

3 EXPERIMENT

3.1 Apparatus

We used a desktop PC (Intel Core i9-12900KF, GeForce RTX 3070 Ti, 32GB RAM, Windows 10 Home). The display was manufactured by AOPEN (model 25XV2QFbmiiprx; 24.5" diagonal, 1920×1080 pixels) and had a 360 Hz refresh rate. The input device was an optical mouse: Logitech G300s gaming mouse (1600 dpi). To match the participant's usual settings, we set the mouse-cursor speed via the OS setting to the middle of the slider in the control-display with "Enhance pointer precision" turned on. The experimental system was implemented with Hot Soup Processor 3.6 and used in full-screen mode.

3.2 Participants

Twelve students from a local university participated in the experiment. The average age was 22.3 years ($SD = 1.92$). All participants were familiar with mouse operation. Experiments were performed with the right hand, which is the dominant hand.

3.3 Task

We created the task for our experiment with reference to experiments 1 and 2 in Oba et al.'s study [19]. Fig. 3 (Left) shows a schematic of the task. A red starting area and a green target were presented on a gray background. When the participant clicked on the starting area, the trial began. The disappearance of the starting area was used as visual feedback to let the participants know that the trial had begun. Participants moved the cursor quickly and accurately toward the target and terminated the trial with the next click. If this click was made within the target, we recorded the trial as a success and presented a pleasant sound to signal the success of the task. If the trial was not successful, we recorded the trial as a failure (error) and presented a beep.

3.4 Design

We set the notch as a black rectangle with a height of 6 mm (22 pixels) and a width of 37 mm (130 pixels) by referring to the actual dimensions of the notch on MacBook Pro (2021). When the cursor overlaps the notch, the overlapping portion of the cursor is hidden by the notch (same specification as the current notch).

Three cursor conditions (*Cursor*) were set: cursor with default size (*default*), cursor twice the default size (*twice*), and cursor warping to the left and right of the notch (*warp*) (Fig 3 (right)). The default size was based on the default actual size of MacBook Pro (2021). Additionally, in *warp*, the cursor size was the default size. The cursor stops at the bottom of the notch, just as it does when the button on the menu bar is selected on the MacBook Pro (2021).

We set two conditions for the cursor movement strategy (*Strategy*), as in a previous study [19]: moving the cursor along the top edge of the screen (*along*) and moving the cursor by avoiding the notch (*avoid*). *Strategy* was used as a condition considering Oba et al. [20], who showed that the strategy of target-aiming operations affects the movement time.

We set the target height to 6 mm, which was the same as the notch, and the width (W) to 6 mm (21 pixels). W was based on the actual size of the target on the taskbar of a MacBook Pro (2021). To avoid increasing the workload on the participants by increasing the number of cases, we used one of the W values used in the previous study [19]. The starting area was always the same size as the target.

We set the movement amplitude (A) from the starting area to the target to two conditions: 100 and 200 mm (350 and 700 pixels, respectively). The value of A was set with reference to the values in a previous study [19].

We set the interval (I) between the notch and target to three conditions: 0, 12, and ∞ mm (0, 42, and ∞ pixels, respectively), with reference to a previous study [19, 22]. Moreover, ∞ mm means no notch condition. The notch is always placed between the targets.

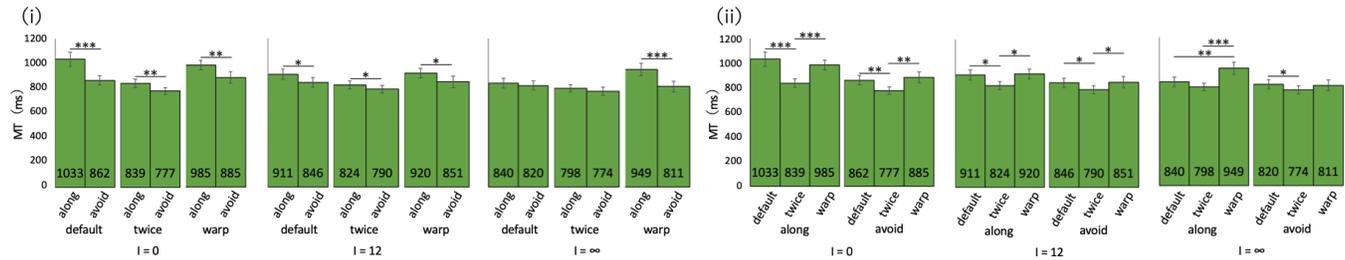


Figure 4: Results of MT of (i) $I \times \text{Cursor} \times \text{Strategy}$, and (ii) $I \times \text{Strategy} \times \text{Cursor}$.

When pointing to a target at the top of the screen, it is known that (i) the effect of the cursor movement direction on the movement time is the same in the left and right directions and (ii) there is no significant difference in the movement time between a typical arrow cursor and a circular cursor [2]. To avoid increasing the workload of the participant, we always placed the starting area to the left of the target. We used the common arrow cursor.

3.5 Procedure

Participants performed the tasks using in two *Strategies* for each *Cursor*. First, the participants were given an explanation of one *Strategy* and practiced 24 trials with one *Cursor*. Then, 20 blocks (120 trials) were performed for data collection, each block comprising a random ordering of 6 trials ($=2A \times 3I$). After completing 20 blocks, the participants took a 1 min break. After the break, participants were explained the other *Strategy*. Then, the participants practiced 24 trials and performed 20 blocks. The order of *Cursor* and *Strategy* was counterbalanced among the participants. Each participant took approximately 30 min for this experiment.

We instructed the participants to (1) perform the task as quickly and accurately as possible, (2) confirm the conditions presented before starting the trial, and (3) avoid clutching (the action of floating the mouse) as much as possible. However, because the distance A in this experiment was sufficient to move the cursor without clutching, the effect of restricting clutching was negligible.

3.6 Measurements

The movement time (MT) is the time from clicking the starting area to clicking the target, excluding error trials. The error rate (ER) is the percentage of tasks recorded as failure out of all tasks.

4 RESULT

We recorded 8,640 ($2(A) \times 3(I) \times 2(\text{Strategy}) \times 3(\text{Cursor}) \times 20(\text{block}) \times 12(\text{participants})$) trails, which included 593 error trails ($ER = 6.86\%$). ER was higher than that in previous studies [16, 19, 20, 24] owing to the small target size used in this experiment, which increased the overall difficulty of the task [10]. We analyzed error-free trials by *mean-of-means* calculation via repeated-measures ANOVA using the Bonferroni method as a p-value adjustment; this is because our analysis was similar to that in previous studies [22, 24]. Error bars in the graphs indicated standard errors. ***, **, and * in the graphs indicated $p < 0.001$, $p < 0.01$, and $p < 0.05$, respectively.

4.1 Error Rate (ER)

We recorded 593 error trials ($ER = 6.86\%$). This ER tends to be higher than that in previous studies [16, 19, 20, 24], which is probably because of the small target size used in this experiment and the overall difficulty of the task [10]. We observed the main effects of *Strategy* ($F_{1,11} = 5.60$, $p < 0.05$, $\eta_p^2 = 0.337$). The other parameters did not show the main effect. No significant interaction was observed. Although *avoid* tended to be higher than *along* for all *Cursor*, significant differences were found only for *warp*.

4.2 Movement Time (MT)

We observed the main effects of A ($F_{1,11} = 157$, $p < 0.001$, $\eta_p^2 = 0.935$), I ($F_{2,22} = 93.8$, $p < 0.001$, $\eta_p^2 = 0.895$), *Strategy* ($F_{1,11} = 22.5$, $p < 0.001$, $\eta_p^2 = 0.672$), and *Cursor* ($F_{2,22} = 20.9$, $p < 0.001$, $\eta_p^2 = 0.655$). Pair-wise comparisons showed that MT for *avoid* was shorter than that for *along* ($p < 0.001$). Furthermore MT increased as A increased ($p < 0.001$), and I decreased ($p < 0.001$ to 0.05 for all pairs). These trends were similar to those observed in a previous study [19, 20, 24]. We observed significant interactions of $A \times \text{Strategy}$ ($F_{1,11} = 6.32$, $p < 0.05$, $\eta_p^2 = 0.365$), $I \times \text{Strategy}$ ($F_{2,22} = 15.6$, $p < 0.001$, $\eta_p^2 = 0.587$), $I \times \text{Cursor}$ ($F_{4,44} = 15.8$, $p < 0.001$, $\eta_p^2 = 0.589$), *Strategy* \times *Cursor* ($F_{2,22} = 3.99$, $p < 0.05$, $\eta_p^2 = 0.266$), and $I \times \text{Strategy} \times \text{Cursor}$ ($F_{4,44} = 10.7$, $p < 0.001$, $\eta_p^2 = 0.492$ (Fig. 4)).

From the significant interaction of $I \times \text{Strategy} \times \text{Cursor}$, although there was a significant difference in *Strategy* for all *Cursor* at $I = 0$ and 12, there was a significant difference only at *warp* at $I = \infty$. At $I = 0$ and 12, *default* and *warp* were significantly longer than *twice*. However, at $I = \infty$, *warp* was significantly longer than *default* and *twice* for *along*.

4.3 Path Efficiency (PE)

In *warp*, PE may show over 100% because the notch width was added to the on-screen travel distance when warping. If warping (left-to-right warping in this experiment) was used to shorten the travel path of the mouse, PE would be significantly high. However, if an unnecessary warp (e.g., a right-to-left warp in this experiment) is performed by mistake, PE will show an unreasonably high value. Therefore, we analyzed 8,006 trials, excluding those wherein unnecessary warping occurred (41 trials) and error trials.

Additionally, we did not add the notch width to the virtual travel distance in this analysis. This method of analysis may make it

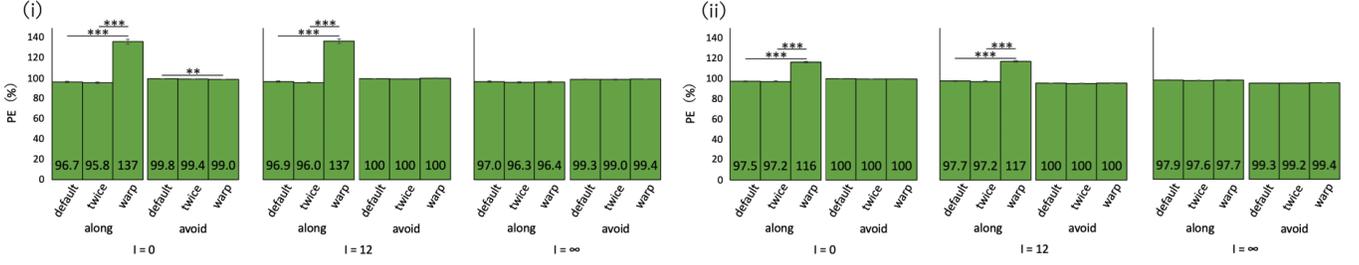
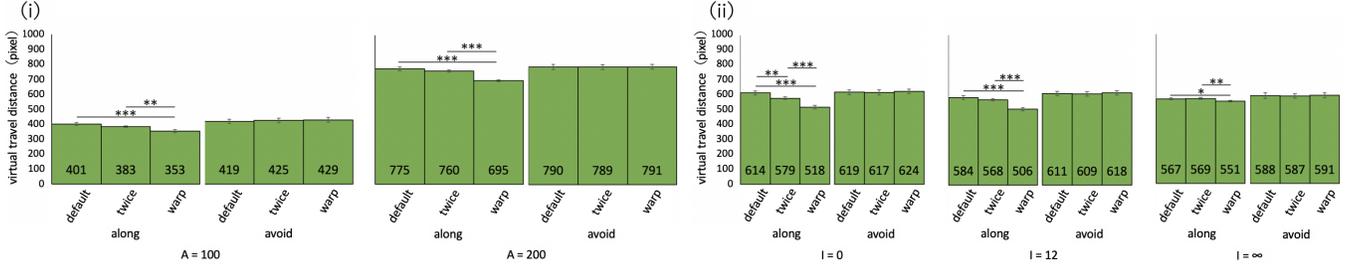
Figure 5: Results of PE of $I \times Strategy \times Cursor$ at (i) $A = 100$ and (ii) $A = 200$.Figure 6: Results of virtual travel distance of (i) $A \times Cursor \times Strategy$ and (ii) $I \times Cursor \times Strategy$.

Table 1: Results of ANOVA for PE.

Factors	DF _{num}	DF _{den}	F	p	η_p^2
A	1	11	46.9	< 0.001	0.810
I	2	22	459	< 0.001	0.977
Strategy	1	11	38.1	< 0.001	0.776
Cursor	2	22	432	< 0.001	0.975
$A \times I$	2	22	161	< 0.001	0.936
$A \times Strategy$	1	11	64.0	< 0.001	0.853
$A \times Cursor$	2	22	178	< 0.001	0.942
$I \times Strategy$	2	22	292	< 0.001	0.964
$I \times Cursor$	4	44	601	< 0.001	0.982
$Strategy \times Cursor$	2	22	580	< 0.001	0.981
$A \times I \times Strategy$	2	22	139	< 0.001	0.927
$A \times I \times Cursor$	4	44	137	< 0.001	0.925
$A \times Strategy \times Cursor$	2	22	137	< 0.001	0.926
$I \times Strategy \times Cursor$	4	44	695	< 0.001	0.984
$A \times I \times Strategy \times Cursor$	4	44	148	< 0.001	0.931

difficult to distinguish between operation with warping and considerable pushing edge and operation with no warping and no pushing edge. However, it is a mistake to add the notch width to the virtual travel distance, because PE is a measure for calculating the efficiency of the cursor's path relative to the mouse travel distance. Therefore, we address this concern by also analyzing the virtual travel distance in PE in the next section.

We present the results of ANOVA for PE in Table 1. In $I = 0, 12$ and *along*, *warp* was significantly higher than *default* and *twice* (Fig. 5).

4.4 Virtual Travel Distance

We analyzed only error-free trials (8,047 trials). We observed the main effects of A ($F_{1,11} = 222 \times 10^2$, $p < 0.001$, $\eta_p^2 = 1.000$), I ($F_{2,22} = 10.0$, $p < 0.001$, $\eta_p^2 = 0.477$), Strategy ($F_{1,11} = 13.3$, $p < 0.01$,

$\eta_p^2 = 0.548$), and Cursor ($F_{2,22} = 31.9$, $p < 0.001$, $\eta_p^2 = 0.744$). We observed significant interactions of $A \times Cursor$ ($F_{2,22} = 5.32$, $p < 0.05$, $\eta_p^2 = 0.326$), $I \times Strategy$ ($F_{2,22} = 23.2$, $p < 0.001$, $\eta_p^2 = 0.678$), $I \times Cursor$ ($F_{4,44} = 12.9$, $p < 0.001$, $\eta_p^2 = 0.539$), $Strategy \times Cursor$ ($F_{2,22} = 26.8$, $p < 0.001$, $\eta_p^2 = 0.709$), $A \times I \times Cursor$ ($F_{4,44} = 3.38$, $p < 0.05$, $\eta_p^2 = 0.235$), $A \times Strategy \times Cursor$ ($F_{2,22} = 9.00$, $p < 0.01$, $\eta_p^2 = 0.450$), and $I \times Strategy \times Cursor$ ($F_{4,44} = 21.0$, $p < 0.001$, $\eta_p^2 = 0.657$ (Fig. 6)). In *along*, *warp* was significantly shorter than other Cursors. In *avoid*, there was no significant difference among Cursors.

5 DISCUSSION

5.1 Movement Time

warp did not show significantly shorter movement time than *default*. The same trend is observed for all Cursors; further, also for *warp*, *avoid* always had significantly shorter movement time than that for *along* (Fig. 4). In contrast, for *along*, *warp* had significantly shorter virtual travel distance than that for the other Cursors (Fig. 6), and PE was significantly higher (Fig. 5). Therefore, we believe that a cursor warping to the left and right of the notch would shorten the distance to move the mouse and improve the efficiency of the cursor path; however, it would not shorten the movement time. Consequently, we found that a cursor warping to the left and right of the notch was ineffective in terms of the movement time.

In *along* and $I = \infty$ (almost similar tasks for all Cursors owing to the no notch condition), *warp* showed significantly longer movement time than the other Cursors. For *warp*, warp was performed on the left and right of the notch for $I = 0, 12$, but not for $I = \infty$, considering there was no notch. Therefore, the cursor behaves differently depending on I, and the distance the participant needs to move the mouse is different even if A is the same. Furthermore, participants

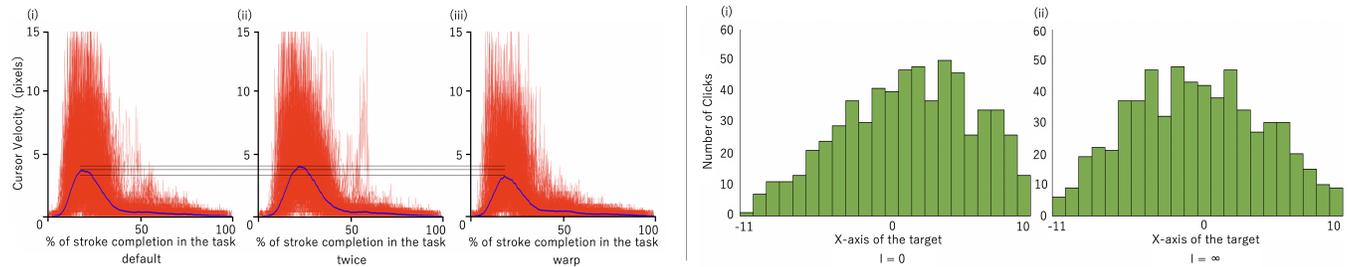


Figure 7: (Left) Cursor velocity against the percentage of stroke completion in the task in (i) *default*, (ii) *twice*, and (iii) *warp*. The red line shows the actual cursor velocity and the blue line shows the average velocity. (Right) Number of clicks made on the X-axis of the target in (i) $I = 0$ and (ii) $I = \infty$.

had to change their mouse operation depending on the presence or absence of warping. The mix of warping and non-warping conditions may contribute to the significantly longer movement time, even in the no notch condition. It may be desirable to change the experimental procedure and do not mix with and without warps in the same block. However, using a cursor warping to the left and right of the notch in real situations results in a mixture of situations in where warping does and does not take place, because the notch is always positioned on the screen. This experiment can help evaluate the performance of a cursor warping to the left and right of the notch, assuming it is used in a real situation.

The *twice* has significantly shorter movement time compared to the other *Cursors*. Additionally, the same trend is not only observed for all *Cursors* but also for *twice*; *avoid* tends to have a shorter movement time than that of *along* (Fig. 4). In other words, changing the cursor size doubling eliminates the situation where the cursor is completely hidden by the notch; however, the cursor is partially hidden, which increases the movement time. The movement time was always shorter for *twice* than that for the other *Cursors*, even for *avoid*. No significant difference was observed in the movement time in I for *twice*. Consequently, we believe that the increased movement time caused by the notch could be restrained by avoiding the notch and using a cursor twice the default size. This result did not contradict the result of the previous study [19]. Therefore, the conclusion that the notch should be an area where the cursor cannot enter, as claimed by the previous study [19], would not be overturned.

5.2 Participants Questionnaire

Ten of the twelve participants answered that they preferred *twice*. They reported that in *along*, the cursor was not completely hidden by the notch and the cursor position was always visible. The large cursor size made it easier to recognize the cursor position and was preferred in *avoid*. The other two participants preferred *default*. For *twice*, the participants highlighted the same advantages as above but felt uncomfortable because the size was larger than the cursor they usually use.

No participant preferred *warp*. The advantage of warping was that the distance to move the cursor was reduced and the cursor was not hidden by the notch. However, they responded that the task became more difficult because the warp caused the cursor to move abruptly. Therefore, the disadvantage of this cursor was

that it required careful moving of the cursor even before warping. Furthermore, it was difficult to aim at the target because of the distance of the warp. Even in *avoid*, the participants stated that they had to move carefully to avoid unnecessary warps from right to left.

5.3 Cursor Velocity

Fig. 7 (left) shows the cursor velocity against the percentage of stroke completion in the task. As an example, $A = 200$, $I = 0$, and *along* conditions are shown. In a pointing task, the user accelerates the cursor at the beginning of the movement time and decelerates to place the cursor within the target [3]. Fig. 7 (left) shows a similar trend in the previous study [3]. However, *warp* (Fig. 7 (left) (iii)) has a smaller maximum average cursor velocity compared to that of the other *Cursors*. The trend was similar for all conditions for *along*. This result supports the answer given in the participant questionnaire. We believe that the participants carefully moving the cursor in *warp* resulted in the decreased movement time.

5.4 Selection Coordinates of the Target

Fig. 7 (right) shows the number of clicks made on the X-axis of the target. As an example, *avoid* and *warp* conditions are shown. The center of the target's X-axis was set to 0, negative to the left and positive to the right. That is, in $I = 0$, -11 was the coordinate 1 pixel to the right of the notch. The number of clicks on a target ideally followed a standard normal distribution [16, 21]. Fig. 7 (right) (ii) shows a trend of a greater number of clicks toward the center of the target. Fig. 7 (Right) (i) shows a trend toward fewer clicks on the left side of the target and more clicks on the right side than in (ii). At $I = 0$, participants moved the cursor to avoid clicking at coordinates close to the notch to avoid unnecessary warping. Furthermore, the participants would have perceived the width of the target at $I = 0$ to be smaller than its actual width.

6 LIMITATION AND FUTURE WORK

Ninja Cursor [15], which shortens the distance to the target, and Bubble Cursor [11], which extends the range of target selection, have been proposed as techniques to support pointing that can reduce the movement time. Therefore, effective use of these techniques may minimize the effect of the notch. A limitation of this study is that existing techniques to support pointing were not included in the cursor conditions.

In the participant questionnaire, they responded that they could have moved the cursor faster if they had become accustomed to *warp* behavior. As the participants become more adept at cursor specifications, they may only benefit from warping to shorten the distance to the target. However, habituation will not eliminate the fact that (1) participants perceive the target width to be smaller than its actual size and (2) there is a mix of situations where warping does and does not take place.

This experiment showed that it is desirable to use a cursor twice the default size. However, we have not been able to investigate how increasing the cursor size affects the everyday use of a PC (e.g., users may feel that a large cursor is annoying). Therefore, we changed the cursor size of the PC to twice the default size and let them use it for one week after the experiment and collected questionnaire responses. All participants reported that they did not lose track of the cursor because of its large size. Conversely, all participants except one said that the cursor shape for selecting text (eye beam pointer) made it difficult to understand where the text could be selected. Four participants continued to use the cursor with twice the size; two changed the size to approximately 1.5 times the default size, and six returned to the default size. Although a cursor twice the default size was desirable in terms of the movement time, we believe that there are individual differences in cursor size preferences. A limitation of this study is that it does not consider the size preference and the effect of cursor size on the pointing movement of the cursor.

7 CONCLUSION

We investigated the possibility of using a cursor warping to the left and right of the notch to shorten the path to the target and restrain the movement time, which increases due to the notch. The results showed that a warping cursor could improve the efficiency of the cursor path, but it was ineffective in terms of the movement time. This result was attributed to a slower in cursor speed, the participants perceiving the target size being smaller than actual, and the mixture of warping and non-warping situations, when using the warping cursor.

ACKNOWLEDGMENTS

We thank the reviewers of CHI 2023 LBW for their valuable feedback.

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* (Ft. Lauderdale, Florida, USA) (CHI '03). Association for Computing Machinery, New York, NY, USA, 193–200. <https://doi.org/10.1145/642611.642646>
- [2] 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* (Napoli, Italy) (AVI '08). Association for Computing Machinery, New York, NY, USA, 119–126. <https://doi.org/10.1145/1385569.1385590>
- [3] Takeshi Asano, Ehud Sharlin, Yoshifumi Kitamura, Kazuki Takashima, and Fumio Kishino. 2005. Predictive Interaction Using the Delphian Desktop. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology* (Seattle, WA, USA) (UIST '05). Association for Computing Machinery, New York, NY, USA, 133–141. <https://doi.org/10.1145/1095034.1095058>
- [4] Patrick Baudisch, Edward Cutrell, Ken Hinckley, and Robert Gruen. 2004. Mouse Ether: Accelerating the Acquisition of Targets across Multi-Monitor Displays. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems* (Vienna, Austria) (CHI EA '04). Association for Computing Machinery, New York, NY, USA, 1379–1382. <https://doi.org/10.1145/985921.986069>
- [5] Renaud Blanch and Michael Ortega. 2011. Benchmarking Pointing Techniques with Distractors: Adding a Density Factor to Fitts' Pointing Paradigm. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) (CHI '11). Association for Computing Machinery, New York, NY, USA, 1629–1638. <https://doi.org/10.1145/1978942.1979180>
- [6] Edward RFW Crossman. 1956. *The measurement of perceptual load in manual operations*. Ph. D. Dissertation. University of Birmingham.
- [7] Coskun Dizmen, Errol R. Hoffmann, and Alan H.S. Chan. 2014. Movement time to edge and non-edge targets. *Ergonomics* 57, 1 (2014), 130–135. <https://doi.org/10.1080/00140139.2013.855824> PMID: 24283932.
- [8] J. Shawn Farris, Keith S. Jones, and 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. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 45, 15 (2001), 1205–1209. <https://doi.org/10.1177/154193120104501511> arXiv:https://doi.org/10.1177/154193120104501511
- [9] Paul M Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology* 47, 6 (1954), 381–391. <https://doi.org/10.1037/h0055392>
- [10] Julien Gori, Olivier Rioul, and Yves Guiard. 2018. Speed-Accuracy Tradeoff: A Formal Information-Theoretic Transmission Scheme (FITTS). *ACM Trans. Comput.-Hum. Interact.* 25, 5, Article 27 (Sept. 2018), 33 pages. <https://doi.org/10.1145/3231595>
- [11] Tovi Grossman and Ravin Balakrishnan. 2005. The Bubble Cursor: Enhancing Target Acquisition by Dynamic Resizing of the Cursor's Activation Area. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Portland, Oregon, USA) (CHI '05). Association for Computing Machinery, New York, NY, USA, 281–290. <https://doi.org/10.1145/1054972.1055012>
- [12] Errol Hoffmann and ILYAS SHEIKH. 1994. Effect of varying target height in a Fitts' movement task. *Ergonomics* 37 (06 1994), 1071–1088. <https://doi.org/10.1080/00140139408963719>
- [13] Stéphane Huot, Olivier Chapuis, and Pierre Dragicevic. 2011. TorusDesktop: Pointing via the Backdoor is Sometimes Shorter. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) (CHI '11). Association for Computing Machinery, New York, NY, USA, 829–838. <https://doi.org/10.1145/1978942.1979064>
- [14] Steven Jax, David Rosenbaum, and Jonathan Vaughan. 2007. Extending Fitts' Law to manual obstacle avoidance. *Experimental brain research. Experimentelle Hirnforschung. Expérimentation cérébrale* 180 (08 2007), 775–9. <https://doi.org/10.1007/s00221-007-0996-y>
- [15] Masatomo Kobayashi and Takeo Igarashi. 2008. Ninja Cursors: Using Multiple Cursors to Assist Target Acquisition on Large Screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) (CHI '08). Association for Computing Machinery, New York, NY, USA, 949–958. <https://doi.org/10.1145/1357054.1357201>
- [16] I. Scott MacKenzie. 1992. Fitts' Law as a Research and Design Tool in Human-Computer Interaction. *Human-Computer Interaction* 7, 1 (1992), 91–139. https://doi.org/10.1207/s15327051hci0701_3
- [17] I. Scott MacKenzie and William Buxton. 1992. Extending Fitts' Law to Two-Dimensional Tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Monterey, California, USA) (CHI '92). Association for Computing Machinery, New York, NY, USA, 219–226. <https://doi.org/10.1145/142750.142794>
- [18] David E Meyer, Richard A Abrams, Sylvan Kornblum, Charles E Wright, and JE Keith Smith. 1988. Optimality in human motor performance: ideal control of rapid aimed movements. *Psychological review* 95, 3 (1988), 340. <https://doi.org/10.1037/0033-295x.95.3.340>
- [19] Yosuke Oba and Homei Miyashita. 2022. Cursor ga sinnyu dekinai notch ga pointhing no sousajikan ni ataeru eikyo (Impact of a notch into which the cursor cannot enter on pointing movement time). *IPSJ SIG Technical Report* 2022-HCI-200 (2022), 1–8. <http://id.nii.ac.jp/1001/00219017/>
- [20] Yosuke Oba and Homei Miyashita. 2022. Notch ga pointhing no sousajikan ni ataeru eikyo (Impact of the notch on pointing movement time). *IPSJ SIG Technical Report* 2022-HCI-199, 2 (2022), 1–8. <http://id.nii.ac.jp/1001/00221974/>
- [21] 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. *Int. J. Hum.-Comput. Stud.* 61, 6 (dec 2004), 751–789. <https://doi.org/10.1016/j.ijhcs.2004.09.001>
- [22] Hiroki Usuba, Shota Yamanaka, and Homei Miyashita. 2021. A Model for Pointing at Targets with Different Clickable and Visual Widths and with Distractors. In *Proceedings of the 32nd Australian Conference on Human-Computer Interaction* (Sydney, NSW, Australia) (OzCHI '20). Association for Computing Machinery, New York, NY, USA, 1–10. <https://doi.org/10.1145/3441000.3441019>
- [23] Jonathan Vaughan, Deborah A Barany, Anthony W Sali, Steven A Jax, and David A Rosenbaum. 2010. Extending Fitts' Law to three-dimensional obstacle-avoidance movements: support for the posture-based motion planning model. *Experimental*

- brain research* 207, 1 (2010), 133–138. <https://doi.org/10.1007/s00221-010-2431-z>
- [24] Shota Yamanaka. 2018. Mouse Cursor Movements towards Targets on the Same Screen Edge. In *Proceedings of the 44th Graphics Interface Conference* (Toronto, Canada) (*GI '18*). Canadian Human-Computer Communications Society, Waterloo, CAN, 115–122. <https://doi.org/10.20380/GI2018.16>
- [25] Shota Yamanaka and Wolfgang Stuerzlinger. 2020. Necessary and Unnecessary Distractor Avoidance Movements Affect User Behaviors in Crossing Operations. *ACM Trans. Comput.-Hum. Interact.* 27, 6, Article 44 (nov 2020), 31 pages. <https://doi.org/10.1145/3418413>
- [26] 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* (Atlanta, Georgia, USA) (*CHI '10*). Association for Computing Machinery, New York, NY, USA, 1391–1400. <https://doi.org/10.1145/1753326.1753536>
- [27] 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* (Austin, Texas, USA) (*CHI '12*). Association for Computing Machinery, New York, NY, USA, 3185–3194. <https://doi.org/10.1145/2207676.2208737>