

Modeling Fully and Partially Constrained Lasso Movements in a Grid of Icons

Shota Yamanaka

Yahoo Japan Corporation
Chiyoda-ku, Tokyo, Japan
syamanak@yahoo-corp.jp

Wolfgang Stuerzlinger

Simon Fraser University
Vancouver, Canada
w.s@sfu.ca

ABSTRACT

Lassoing objects is a basic function in illustration software and presentation tools. Yet, for many common object arrangements lassoing is sometimes time-consuming to perform and requires precise pen operation. In this work, we studied lassoing movements in a grid of objects similar to icons. We propose a quantitative model to predict the time to lasso such objects depending on the margins between icons, their sizes, and layout, which all affect the number of stopping and crossing movements. Results of two experiments showed that our models predict fully and partially constrained movements with high accuracy. We also analyzed the speed profiles and pen stroke trajectories and identified deeper insights into user behaviors, such as that an unconstrained area can induce higher movement speeds even in preceding path segments.

CCS CONCEPTS

• **Human-centered computing** → **HCI theory, concepts and models**; **Pointing**; **Empirical studies in HCI**;

KEYWORDS

Graphical User Interfaces; Human Motor Performance; Lassoing; Steering

ACM Reference Format:

Shota Yamanaka and Wolfgang Stuerzlinger. 2019. Modeling Fully and Partially Constrained Lasso Movements in a Grid of Icons. In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019)*, May 4–9, 2019, Glasgow, Scotland UK. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3290605.3300350>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CHI 2019, May 4–9, 2019, Glasgow, Scotland UK

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-5970-2/19/05...\$15.00

<https://doi.org/10.1145/3290605.3300350>

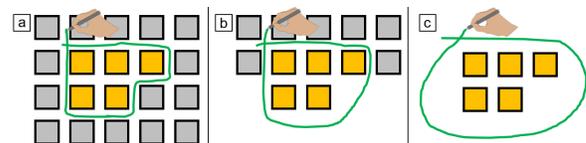


Figure 1: Different stroke constraints for lassoing movements. The target group in yellow needs to be selected by passing through the white areas. Illustration of a (a) fully constrained, (b) partially constrained, and (c) unconstrained stroke. Here we target conditions (a) and (b). The lasso criterion here is the same as Figure 3d.

1 INTRODUCTION

Selecting multiple objects is a basic operation in graphical user interface (GUI) applications, such as presentation tools and illustration software. Widely used methods for selecting object groups include individually tapping the intended items (while holding a modifier key down) or drawing a selection stroke to include the intended icon group in the corresponding rectangle. Another common way is *lassoing* – a continuous stroke around the intended items, as shown in Figure 1. As lassoing can be time-consuming, especially for spatially large icon groups [7], researchers have experimented with improvements, such as automatically connecting the start and end points [25] and considering the likelihood of objects belonging to the same group to facilitate selection [9].

To develop novel techniques for such a task, it is important to establish a baseline model. For example, new mouse pointing techniques that reduce the movement distance (A) and/or increase the target size (W) have been proposed [15, 16] based on Fitts’ law [12]. With a robust model, the effectiveness of a new technique can then be predicted, which lessens the need to conduct user studies. However, there is little work on modeling operation times for lassoing movements for given task parameters. For example, when the icon size and margins are both 150% larger than the default condition of Figure 1a, how does the task completion time change?

Building on existing motor performance models we derive here a model to predict movement times for lassoing tasks for conditions where the objects to be selected are square-shaped and arranged in a grid. Because we are interested in establishing a baseline model, we ignore supporting lasso techniques,

such as auto-closing the stroke, and focus our work on layouts with a grid of icons. Lassoing completely unconstrained icon groups (Figure 1c) has been investigated before [13, 36]. Thus, we focus on fully- or partially-constrained conditions (Figure 1a and b). Our contributions include:

- **Model development based on existing performance models.** We present a model to predict a movement time for a lassoing task parameterized by the margin between icons (or path width W), icon size (S), and layouts of icons that affect the number of corners and lengths of straight path segments.
- **Validation of candidate models via two experiments.** The results of two experiments show a high model fitness for fully- and partially-constrained conditions through adjusted R^2 and AIC [5]. Our work demonstrates that operation times can be estimated by summing each difficulty by careful segmentation of required motions, even for a complex object lassoing task.
- **Analyses of speed profiles and trajectories to deepen understanding of user behaviors.** We also illustrate that for the same movement distance and path width, the layout can significantly affect performance. Our speed profile and pen trajectory analysis reveals that users consistently change their movement behavior to anticipate future corners and “gates” to be entered.

2 RELATED WORK

Selection Methods for Multiple Objects

Clicks on target objects and rectangle selection are frequently supported mechanisms to select multiple objects. For example, in Windows, holding the Ctrl-key down and clicking on intended objects adds them to the group. Tapping on selected item removes them from the group. A (diagonal) stroke through a group is usually mapped to rectangle selection, which selects all objects within the rectangle’s area.

Brushing, a one-dimensional selection technique, is another way to select multiple items; users make a stroke that passes through the intended items. To avoid inadvertent selection of items that are just touched by the stroke, smart algorithms adjust the selection [38], but this approach is not always easy to understand for users [7].

A lassoing tool, also called *free selection* in drawing software (e.g., Gimp), is efficient for selecting a complex-shaped group with many items. For small groups, just clicking on each item is faster than making a stroke around. Also, the combination of rectangle selection with a few additional taps (adding or removing some objects) can be efficient [33].

The baseline performance of lasso operations has been measured, typically in comparisons with novel techniques [7–9, 25, 38]. Yet, no general model exists for lassoing performance, even for simple task, such as in a grid of icons.

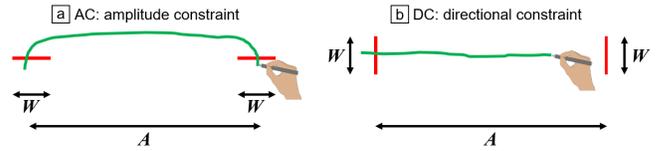


Figure 2: Continuous crossing tasks [4, 6].

Bjerre et al. proposed a model linear in the number of lassoed icons [7], which is overly simplistic. Their result can be explained by their choice of target groups. Also, their work does not take different icon sizes, number of corners, or free areas into account. Thus, the effects of such task characteristics on lassoing are unexplored.

Performance Models for GUI Operations

Pointing Model. Fitts’ law [12] in the Shannon formulation [24, 32] predicts movement time for aiming tasks:

$$MT = a_p + b_p ID_p, \quad ID_p = \log_2(A/W + 1) \quad (1)$$

where MT is the time to point to a target, A is the distance, W is the size of the target, and a_p and b_p are empirically determined constants. The logarithmic term is called the index of difficulty of pointing (ID_p).

Selecting multiple items by tapping is accurately modeled by adding multiple Fitts’ law terms [18]. Thus, the time for adding (or deleting) individual items to (from) a group after a lasso selection is easily predicted.

Crossing Model. A crossing operation, as shown in Figure 2, is also well modeled by Fitts’ law [4, 6]: passing through a line of length W at a distance A is modeled by: $MT = a_c + b_c ID_c$, where $ID_c = \log_2(A/W + 1)$ and a_c and b_c are crossing regression constants. Targets both collinear and perpendicular to the movement direction, see Figure 2a and b, are supported, with slightly different a_c and b_c ’s [4, 6]. This model also predict the time to enter a “gate” between obstacles around a path [41, 42]. We similarly assume that moving the pen from an unconstrained into a constrained area during lassoing is modeled by the crossing law, through a term that describes the task geometry for such a transition.

Steering Model. Models for steering through a constrained path have been proposed for car driving [30] and pen drawing [11]. For GUI operations, Accot and Zhai [1] proposed a global model to steer through a path (or tunnel) T :

$$MT = a_s + b_s ID_s, \quad ID_s = \int_T \frac{dx}{W(x)} \quad (2)$$

where x is the cursor position, $W(x)$ is the path tolerance width at x , and a_s and b_s are empirically determined steering constants. The integral term is called the steering index of difficulty ID_s . If the width is constant throughout the path, the model simplifies to $MT = a_s + b_s(A/W)$, where A is the

path length, and W the path width. This relationship holds for various devices [2] and movement angles [34, 35].

Model of Steering with Cornering. Here, we discuss several models that model a part of lasso movements. First, an entire lassoing motion could be directly modeled by the global steering model (Equation 2). That is, if the path and its width at every point are known then the MT could be predicted. However, previous work found evidence to the contrary.

Pastel’s work on steering tasks with a corner [29] shows that users decelerate when approaching the corner, stop there, and then accelerate in the second path segment. As the steering model assumes a continuous movement with speed linearly related to the path width W [1, 11, 17], such a stop-and-go motion is not modeled well. Pastel thus regarded this stop as a pointing motion, and modeled as follows:

$$MT = a + b_s \frac{A}{W} + b_p \log_2 \left(\frac{A/2}{W} + 1 \right) \quad (3)$$

where A is the total path length, and W the path width. The total movement time is then the sum of the times estimated by the global steering model and a Fitts’ law term for stopping at the corner. As there is only a single corner between two path segments of the same length, the constant a includes both intercepts for steering (a_s) and pointing (a_p). Also, because there are two steering motions, the intercept of a_s is a merged value of $a_s \times 2$. Thus, we rewrite Equation 3:

$$MT = \left[a_s + b_s \frac{A_1}{W} \right] + \left[a_p + b_p \log_2 \left(\frac{A_1}{W} + 1 \right) \right] + \left[a_s + b_s \frac{A_2}{W} \right] \quad (4)$$

where A_1 and A_2 are lengths of the first and second path segments, and the terms model the first steering, middle pointing, and final steering task, respectively.

Model of Steering followed by Pointing. When users select an item in a hierarchical menu, they steer through a parent menu and then click on the intended command in the child menus. Such an operation is called a *targeted steering motion* [20, 21], which has been modeled several times [10, 21, 31]. Dennerlein et al.’s model uses ID_s and ID_p and merges a_s and a_p into a parameter a : $MT = a + b_s(ID_s) + b_p(ID_p)$, which showed an excellent fit, $R^2 > 0.98$. To reduce the number of parameters Kulikov and Stuerzlinger [21] further combined the ID_s and ID_p terms, $MT = a + b(ID_s + ID_p)$, but model fitness decreased because the slopes of steering and pointing (b_s and b_p) are different [31]. Senanayake et al. proposed to separate steering and pointing phases [31], but found that Dennerlein et al.’s model still fitted the data well, too.

Model of Sequential Steering Tasks. Yamanaka et al. showed that steering through two sequential path segments is accurately predicted by the global steering model [41]. Entering the second path segment is well described as a crossing task.

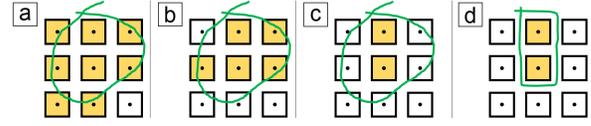


Figure 3: Inclusion criteria for objects in a lasso. Yellow icons are selected by the green stroke. (a) Objects fully inside or crossed by the stroke, (b) objects whose center is inside [25], (c) objects whose entire area is inside, and (d) objects whose entire area is inside of a stroke that passes only through empty areas. Here we use the last criterion.

For lasso movements, we will thus use the crossing law (ID_c) to model the task of entering a constrained area.

Models of Gesturing. Gestures, used, e.g., for marking menus [22], entering symbols via a recognizer [37], or shapewriting [44], are performed as a free-form stroke. The task result depends mainly on the closeness between the input gesture and the via-points or symbol templates. Thus, stroke variability is natural and accepted. In contrast, for lassoing the main task requirement is to avoid the inclusions of unintended objects, i.e., constraints external to the stroke. This difference between a given or resultant variability affects the choice of model. For example, the steering model is suitable when a circular path has a limited tolerance width [2, 3], but the minimum-jerk model is appropriate when users draw a circle on an empty screen [36]. To accurately model lassoing, we thus build on models for constrained tasks, i.e., steering, Fitts’, and crossing models.

Criterion for Inclusion of Icons in a Lasso

There are several decision criteria for objects to be included in a lasso stroke. The most common options are shown in Figure 3. The visual difference between these criteria could affect users’ perception and operation speed. Yet, regardless of whichever criterion is used, participants get familiar with a given criterion after sufficient practice. Therefore, we assumed that the choice of lasso criterion would not strongly affect our experimental result. Here, to eliminate ambiguity for users, we match the visual appearance of the path to the task requirement of a tolerance W (Figure 3d). This allows participants to simply steer through white path areas, and evaluating this lasso method yields the highest internal validity, due to the clearly defined tunnel. User performance under Figure 3a–c can be recalculated by changing the task parameter of W .

3 EXPERIMENT 1: FULLY CONSTRAINED PATHS

In this experiment, the path was always fully constrained in a grid layout of icons with margins (W). We assume that in such tasks continuous visual feedback is required to accurately steer the pen-tip through the icons [35, 42], which

makes this a suitable test of how the task parameters affect participants' behaviors. The objective of this experiment was to identify potential steering behavior changes depending on the task conditions, such as the effect of path segment length, for fully constrained lasso movements.

Participants and Apparatus

Twelve participants were recruited from a local university (one female, $M = 22.5$ years, $SD = 1.19$). All had normal or corrected-to-normal vision and were right-handed. Only one participant used pen tablets daily for three years. Each participant received 27 US\$ for their time.

We used a Sony Vaio Z tablet PC (3.1 GHz i7-5557U; 16 GB; Windows 10). The display is 13.3", 293.5×165.0 mm, at 2560×1440 pixels, 0.115 mm/pixel, with 60 Hz refresh rate. The system reads and processes input about 125 times per second. The tablet was positioned flat on a table. The default digitizer stylus pen of the tablet PC was used (13.9 cm; 20 g). The touch display rejected finger touches when the pen-tip was on the surface and participants were informed that their palm could touch the display. The experimental system was implemented with Hot Soup Processor 3.4 and used in full-screen mode.

Task

The task was to make a single, clockwise stroke, which includes only the orange icons, starting at the blue start area. We required participants to cross the start of the stroke, as shown in Figure 4. The current position of the pen was shown with a cross-hair cursor. Participants were not allowed to "touch" any icons, so the pen-tip could only pass through white paths. If an icon was touched or the pen lifted up (including low pen pressure or tilting the stylus too much), a beep sounded and the trial had to be re-done. When the pen tip crossed the stroke itself and all intended icons (only the orange ones) were included in the loop, a bell sound was played to signal success. Participants were asked to make a stroke as quickly and accurately as possible.

Design and Procedure

Margin and Icon Size. We assume that, naturally, the total distance A that the cursor travels affects movement time MT . Based on previous work, including work on passing through successive obstacle pairs [42], we also assume that the tunnel width W and icon size S affect MT . Hence, we varied W and S , and thus the total distance A changed depending on W and S . We tested two values of W (5 and 8 mm, or 44 and 70 pixels, respectively) and two values of S (8 and 11 mm, or 70 and 96 pixels, respectively).

Icon Layout. Because the number of corners [29] and the length of straight path segments [35] affect MT , we also

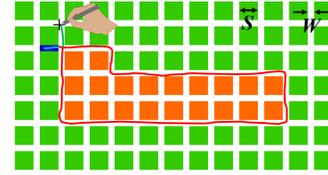


Figure 4: Lassoing operation in Experiment 1. The pen left a trajectory in different colors: light blue in the start area, red in the path, and green after crossing the stroke. MT and steering errors were measured only for the core path (red).

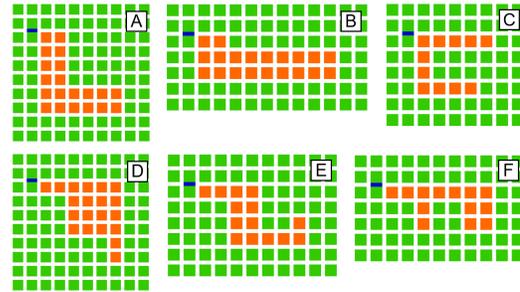


Figure 5: Icon layouts used in Experiment 1.

wanted to analyze the effect of icon layout and thus path shape. Therefore, we tested several layouts under the same conditions on margin W and icon size S (and thus A), shown in Figure 5. Initially, we experimented with more than 50 icon arrangement patterns but ultimately chose six specific layouts with the same outer circumferential distance of 24 icons. The three pairs of layouts (A and B , C and D , and E and F) have the same numbers of corners (5, 7, and 9, respectively), in different arrangements. This enabled us to fairly analyze the effect of the number of corners and straight segments on MT and to confirm that our model is accurate for these conditions. With these stimuli, observing different MT values can then be attributed to the layouts and their properties, such as the numbers of corners and the length of the straight segments where the user can reach the maximum speed [31, 35].

In all the layouts, the orange icons had to be lassoed with a single stroke; we did not investigate layouts that have isolated "islands". Such cases can be modeled through individual models together with appropriate Fitts' law terms [18]. Orange icons were always arranged so that their bounding box was located at the center of the display.

From the 24 total combinations ($2S \times 2W \times 6$ layouts), 10 conditions were randomly selected as practice trials. After that, each participant performed 5 repetitions of the 24 conditions (=120 trials). In total, across all 12 participants we recorded 1,440 trials.

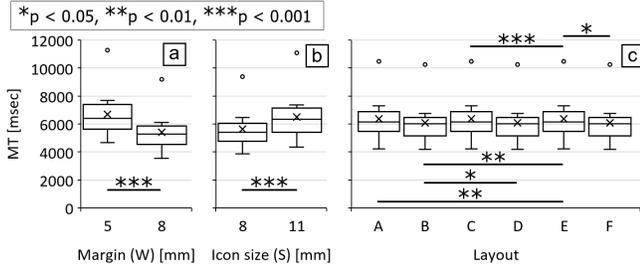


Figure 6: Movement times observed in Experiment 1.

We informed the participants that the system automatically recorded the time from when the cursor left the blue start area to when the cursor crossed the stroke trajectory. Movement direction was always clockwise, hence we asked participants to move rightwards after leaving the start area. Because the whole experiment took about half an hour, we let participants rest whenever they felt fatigue.

Results

We observed 253 retrials, of which 199 (11.8%) were steering errors and 54 (3.2%) pen lifts. Due to the length of the path and its complexity, it is not surprising that the steering error rate was higher than in comparable work [41].

As in previous work [3, 39, 41, 43], we analyzed the MT data of error-free (successful) trials using repeated-measures ANOVA with a Bonferroni post-hoc test.

Movement Time. Figure 6 shows the results of MT . We found significant main effects of the margin W ($F_{1,11} = 123.444$, $p < 0.001$, $\eta_p^2 = 0.918$), icon size S ($F_{1,11} = 85.792$, $p < 0.001$, $\eta_p^2 = 0.886$), and layout ($F_{5,55} = 12.625$, $p < 0.001$, $\eta_p^2 = 0.534$). If we take the number of corners as a dependent variable (i.e., merging the data of *layout-A* and *-B*, *layout-C* and *-D*, and *layout-E* and *-F*), we also identify a significant main effect ($F_{2,22} = 14.798$, $p < 0.001$, $\eta_p^2 = 0.574$).

Speed Measurement using Checkpoint Approach. To fairly compare movement speed profiles between different layouts, we define “check-points” as follows. For each midpoint (crossroad) between two successive icons along the path, a check-point perpendicular to the movement direction was used for analysis, depicted as blue lines in Figure 7a. The speed at each check-point is then calculated as the distance from the previous check-point divided by the time spent for the distance. Here, we assume that the cursor passes through the tunnel on average along the center of the path. This is a reasonable assumption, based on the derivation of the global steering model (see Figure 3 by Drury [11] and Montazer and Drury [26]). The movement distance for each segment is then $(S + W)$, except for the first check-point from the right edge of the start area being $0.5W$. That first check-point is crossed

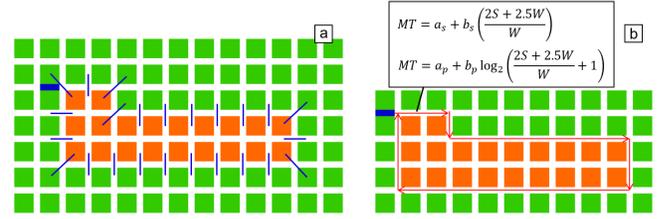


Figure 7: (a) Check-points to measure the movement speed in *layout-B*, and (b) the MT values predicted by the steering and Fitts’ law for the first path segment in Experiment 1.

twice, thus the number of crossings is 25. Examples of speed profiles in *layout-A* and *layout-B* are shown in Figure 8.

Model Derivation and Fitting. Here, we evaluate three models, as shown in Table 1. First, the global steering model (Equation 2), which considers a lassoing task as a single steering motion, with the only explanatory variable being the $ID_s = A/W$ where $A = 24S + 24.5W$ and with a single steering intercept a_s :

$$MT = a_s + b_s \frac{24S + 24.5W}{W} \quad (5)$$

Second, we test the rewritten version of Pastel’s model, called “Steering (segmented)” model (Equation 4), assuming that users perform individual steering motion for each straight path segment. Therefore, the numbers of steering intercept (a_s) for *layout-A* and *B* is 6, for *C* and *D* is 8, and for *E* and *F* is 10. In the second row of Table 1, we use “#s” to show the number of steering motions. The total steering difficulty (ID_s) is the sum of difficulties for each straight path segment. For example, in *layout-B*, the length of the first segment is $2S + 2.5W$, and thus the steering difficulty in that segment is $(2S + 2.5W)/W$. Similarly, the difficulty for the second path segment is $(1S + 1W)/W$, and that for third segment is $(7S + 7W)/W$, and so on. In this manner, the “Steering (segmented)” model for *layout-B* is the sum of the MT values of the six steering motions:

$$MT = \left[a_s + b_s \frac{2S + 2.5W}{W} \right] + \left[a_s + b_s \frac{1(S + W)}{W} \right] + \left[a_s + b_s \frac{7(S + W)}{W} \right] + \left[a_s + b_s \frac{2(S + W)}{W} \right] + \left[a_s + b_s \frac{9(S + W)}{W} \right] + \left[a_s + b_s \frac{3(S + W)}{W} \right] \quad (6)$$

In Table 1, we simply write this as $MT = a_s(\#s) + b_s(ID_s)$.

Third, we test our new model, where a motion in a straight path segment consists of both steering and pointing motions. For example, in *layout-B*, the (straight) steering difficulties are the same as Steering (segmented) model. Yet, as illustrated in Figure 7b, pointing to the first corner is modeled as $MT = a_p + b_p \log_2((2S + 2.5W)/W + 1)$. Then, the total MT

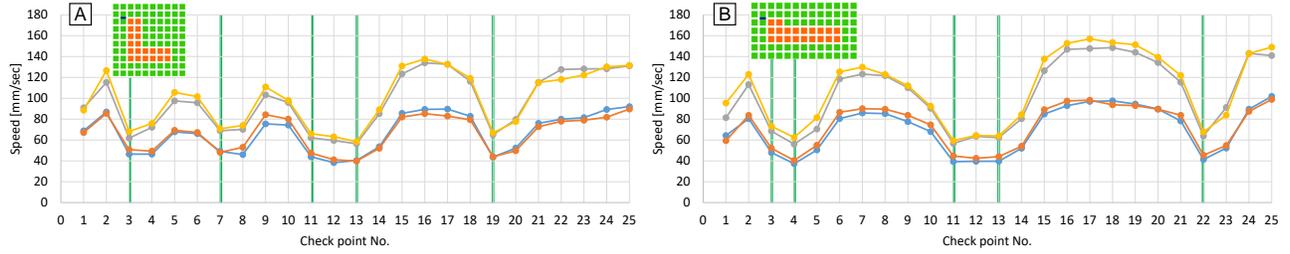


Figure 8: Speed profiles of *layout-A* and *layout-B* in Experiment 1. Green vertical lines indicate corners.

for the five pointing motions is:

$$\begin{aligned}
 MT = & \left[a_p + b_p \log_2 \left(\frac{2S + 2.5W}{W} + 1 \right) \right] + \left[a_p + b_p \log_2 \left(\frac{1(S + W)}{W} + 1 \right) \right] \\
 & + \left[a_p + b_p \log_2 \left(\frac{7(S + W)}{W} + 1 \right) \right] + \left[a_p + b_p \log_2 \left(\frac{2(S + W)}{W} + 1 \right) \right] \\
 & + \left[a_p + b_p \log_2 \left(\frac{9(S + W)}{W} + 1 \right) \right] \quad (7)
 \end{aligned}$$

Note that a pointing motion is not required for the final path segment. Thus, the number of intercepts of pointing ($\#p$) is ($\#s - 1$). In the third row of Table 1, we rewrite the sum of MT s for six steering and five pointing motions as:

$$MT = a_s(\#s) + b_s(ID_s) + a_p(\#p) + b_p(ID_p) \quad (8)$$

Discussion of Experiment 1

Model Fitting. The fit of the global steering model was, somewhat surprisingly, good, with adjusted $R^2 = 0.927$. This indicates a certain robustness of this steering model. A potential explanation is that participants did not perform the task with ideal “stop-and-go” movements at corners [29]. Instead, participants decelerated to a minimum of ~ 40 mm/sec, see Figure 8. Thus the global steering model can still capture a reasonable part of the entire lassoing performance.

Yet, modeling a lasso motion as a sum of steering motions for each straight path segment improved the fitness and AIC values. Hence, if a lassoing task can be divided into successive steering motions, it is better to separate them. Yet, we also show that taking the pointing tasks at corners into account significantly improves fitness, and our proposed model achieved the best scores, with significantly better AIC [5]. As another check for the prediction error, we calculated the SD s of the difference between the predicted and observed movement times ($MT_{\text{predicted}} - MT_{\text{observed}}$) for the $N = 24$ data points (AIC also uses these differences in its calculation). The results are 203, 207, and respectively 129 msec, for the rows of Table 1, which is another indication that we can estimate the MT most accurately with our proposed model.

Speed Profile. The effect of the path width is clearly visible in the speed profiles, see Figure 8. Overall, participants accelerated in straight segments and decelerated in advance of a corner. As expected, speeds at corners were lower to safely

“make” the curve and speeds in the middle of straight segments show peak values. Due to the lack of a final stopping area, speed increased gradually at the trial end, as observed in other steering studies [31, 41, 42].

In short segments the speed could not reach to the potential maximum value, due to the path width [31, 34, 35]. Longer path segment showed a higher average movement speed. For example, participants achieved a higher speed in the longest bottom segment in *layout-B* than in *layout-A* (Figure 8). Another indication for the influence of long straight segments is that the icon size S has a significant effect, as with larger icons, straight segments are longer. This is visible in Figure 8, where the yellow and orange lines ($S = 11$ mm) are slightly higher than the gray and blue ones ($S = 8$ mm).

4 EXPERIMENT 2: PARTIALLY CONSTRAINED PATH

The main objective of Experiment 2 was to explore the effect of unconstrained segments on lassoing. Thus, we compared both fully- and partially-constrained movements.

Participants and Apparatus

Twelve participants were recruited from a local university (one female, $M = 22.8$ years, $SD = 1.74$). They had normal or corrected-to-normal vision and were right-handed. One participant used a direct input pen tablet for one year daily. Three participants had participated in Experiment 1. Each participant received 27 US\$ for their time.

The same PC in Experiment 1 (Sony Vaio Z) was used. To allow participants to accelerate more in large unconstrained areas, we used a Wacom Cintiq 27" QHD pen & touch tablet (2560×1440 pixels, 596.7×335.6 mm active input area, 0.233 mm/pixel; 12 ms response time), and its refresh rate was set at 60 Hz. The system reads and processes input about 125 times per second. The tablet was positioned on a table in “stand” mode (at 20°). Participants were informed that their palms could touch the surface, as we had disabled touch sensing. The digitizer pen was the default accessory of the tablet (24 cm; 19 g). The experimental system was implemented with Hot Soup Processor 3.4 and used in full-screen mode.

Table 1: Model fitting results with 95% CIs [lower, upper] for Experiment 1. Higher adjusted R^2 values are better, while lower AICs are better. #s and #p indicate the numbers of steering and pointing motions, respectively.

Model	Formulation	a_s	b_s	a_p	b_p	adj. R^2	AIC
Steering (global model)	$MT = a_s + b_s(ID_s)$	1428 [858, 1997]	75.7 [66.6, 84.9]			0.927	333
Steering (segmented)	$MT = a_s(\#s) + b_s(ID_s)$	115 [71.1, 158]	83.8 [78.1, 89.5]			0.953	330
Proposed	$MT = a_s(\#s) + b_s(ID_s) + a_p(\#p) + b_s(ID_p)$	1360 [863, 1857]	50.8 [37.1, 64.5]	-1729 [-2371, -1088]	210 [106, 315]	0.949	311

Task and Icon Layout

The task was similar to Experiment 1. We reused some layouts, but deleted parts of the grid to create unconstrained regions. As shown in Figure 9, we reused *layout-B, C, and E*. We attached a suffix of *1* to these fully-constrained conditions, e.g., “*layout-B1*”, to indicate these as baseline layouts. We tested two additional, partially unconstrained conditions for each baseline condition as follows.

With *layout-B2* and *B3*, we aimed to test the effect of the length of an unconstrained segment in a lasso path. With *layout-C2* and *C3*, which both have an unconstrained path along four icons, we wanted to test the effect of the presence/absence of a corner in unconstrained segments. If the angle of the start of the following constrained segment (here, 90° and 0° , respectively) affects *MT*, this would indicate that models need to take this factor into account.

The *layout-E2* includes two unconstrained regions, but the pen movement direction has to change quickly in these regions. Therefore, we assumed that the accelerations in these unconstrained areas were limited, and there would be less of a difference with *layout-E1*. In *layout-E3*, a large curved stroke is needed to maintain a high speed in the unconstrained area. Therefore, we assumed that the total travel distance would increase, compared with *layout-E1* and *E2*, but that the pen speed would be higher.

A major goal of this experiment was to observe the differences in cursor trajectories and speeds, and to test whether a single model showed a good fit regardless of the different conditions. Although the ballistic movement speed in unconstrained area can differ depending on the movement direction [14, 34], the effect of direction would be small relative to other ones, such as the unconstrained distance.

Design, Procedure, and Instruction

The total number of parameter combinations was 36 ($2S \times 2W \times 9$ layouts). 10 conditions were randomly selected as practice trials. After that, participant performed 3 repetitions of the 36 conditions, thus 108 actual trial. In total, for all 12

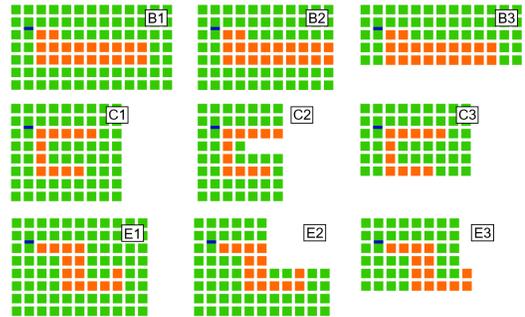


Figure 9: Icon layouts used in Experiment 2.

participants we recorded 1,296 data points for *MT* and errors. This study took 20 to 30 min per participant.

Because we observed a high error rate in Experiment 1, we used larger W values (7 and 11 mm, or 30 and 47 pixels, respectively). We also chose to include longer straight segments ($S = 15$ and 20 mm, or 64 and 86 pixels, respectively). Initially, we assumed that these larger W values would be enough to lower the error rate, e.g., to $<10\%$. However, we observed very high steering error rates of 34% and 47% for two participants in a pilot. Therefore, we decided to instruct the participants to “move quickly as long as you do not touch icons”, which emphasized accuracy over speed. Such an instruction is also common for error-prone tasks, e.g., in Accot and Zhai’s experiment [2]. We did not use even larger W values, because lassoing movements in such conditions would be more ballistic and not require precise pen operation.

Results

We observed 96 retrials, where 82 (=6.3%) were steering errors and 14 pen lift errors (=1.1%).

Movement Time. We found a significant main effects of the margin W ($F_{1,11} = 760.61, p < 0.001, \eta_p^2 = 0.874$), icon size S ($F_{1,11} = 243.38, p < 0.001, \eta_p^2 = 0.954$), and layout ($F_{5,55} = 49.994, p < 0.001, \eta_p^2 = 0.820$), as shown in Figure 10.

Cursor Trajectory. To compare the participants’ behaviors with the different layouts, we overlaid successful trials for

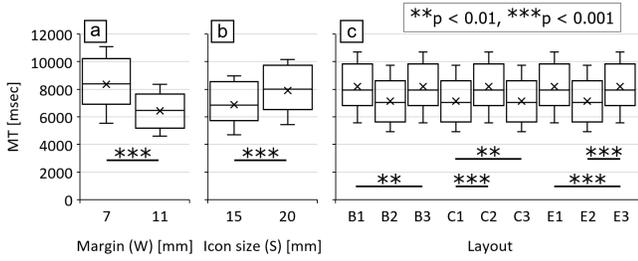


Figure 10: Movement times in Experiment 2. Significant differences between each layout group of (B1, B2, B3), (C1, C2, C3), and (E1, E2, E3), are annotated to compare behaviors with unconstrained segments among these groups.

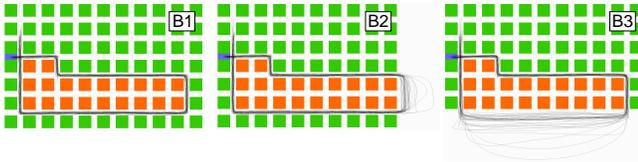


Figure 11: Overlaid cursor trajectories of all successful trials for $W = 7$ mm and $S = 15$ mm in layout-B1, B2, and B3. 48 strokes per image.

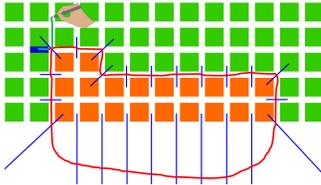


Figure 12: Checkpoints for speed calculation in Exp. 2.

the exemplary condition of $W = 7$ mm and $S = 15$ mm in layout-B1, B2, and B3 (Figure 11). These images clearly show that participants made more sloppy, curved strokes in longer unconstrained areas.

Speed Profile. We again use the same “check-point” method used in Experiment 1 to compare movement speeds. In Experiment 2, as shown in Figure 11, strokes tended to use a more “roundabout” way in unconstrained areas. Thus, check-points in unconstrained areas need to have semi-infinite lengths (Figure 12). The speed at each check-point is calculated with the same method: $S + W$ divided by the time spent for the distance. Specifically in unconstrained areas, actual paths are often longer than $(S + W)$ due to roundabout movements. Yet, such *fast but long* movement do not contribute to progress the lasso task. Thus, regardless of the stroke shape, we define $(S + W)$ as the movement distance for a single segment. The profiles in Figure 13 fairly compare the speeds between check-points within the same layout group.

Model Derivation and Fitting. We again evaluated different models, see Table 2. The path tolerance for unconstrained areas is in essence infinite. For such situations, the global steering model predicts an ID_s of zero. Lank and Sound proposed to limit W via the instantaneous stroke speed observed in unconstrained areas (Equation 10 in [23]). In contrast, our work aims to derive a predictive model that can estimate MT based (only) on the given icon geometry.

For example, in layout-B3, there are 24 outer icons, but nine of them are along the bottom unconstrained area. Thus, the steering difficulty for the bottom unconstrained area is zero, and the ID_s for the global and segmented steering models is $(15S + 15.5W)/W$. Similarly, because there needs no steering motion in the bottom unconstrained area, the number of steering motion ($\#s$) in layout-B3 is 5. In this manner, we obtain the Steering (global) and Steering (segmented) models (first and second rows in Table 2, respectively). Yet, as the speed in unconstrained areas is not actually infinite [28], even when $W = \infty$, these two models do not show good fits. Thus, the global steering model cannot predict the task in Experiment 2 well.

Thus, similar to Experiment 1, we evaluate the fitness of a model of segmented steering with pointing (third row), to check the effectiveness of adding pointing motions for corners via ID_p . In layout-B3, exiting the fourth path segment from constrained to unconstrained areas and entering the final path segment from the unconstrained area need no pointing motions, in addition to the final path segment. Hence, the number of pointing motions ($\#p$) is only 3.

We also evaluate the Steering (segmented) model with a crossing difficulty for transitions from an unconstrained area to a constrained one (fourth row). Although crossing motions for amplitude and directional constraints have slightly different constants (a_c and b_c) [4, 6], we approximate them through a single parameter. In layout-B3, entering the final segment from the bottom unconstrained area is modeled as:

$$MT = a_c + b_c \log_2 \left(\frac{9(S + W)}{W} + 1 \right) \quad (9)$$

For the crossing motion in layout-C2, depicted in Figure 14a, we refer to Hoffmann and Sheikh’s work on pointing with obstacle avoidance [19]. They found that the MT for pointing at a target (closed-loop operation) after avoiding an obstacle (ballistic operation), which requires an additional vertical (height) movement, can be modeled by a Fitts’ law term that uses the sum of each movement distance, see Figure 14b. Users performed the obstacle avoidance and homing-in movements as a single sweeping motion. Their work uses the original Fitts’ law formulation $MT = a + bID$ with $ID = \log_2((2(H + A)/W)$. We adopt this result to a crossing motion for layout-C2. The movement distance is, as shown by the blue line in Figure 14c, $4(S + W)$. Note that

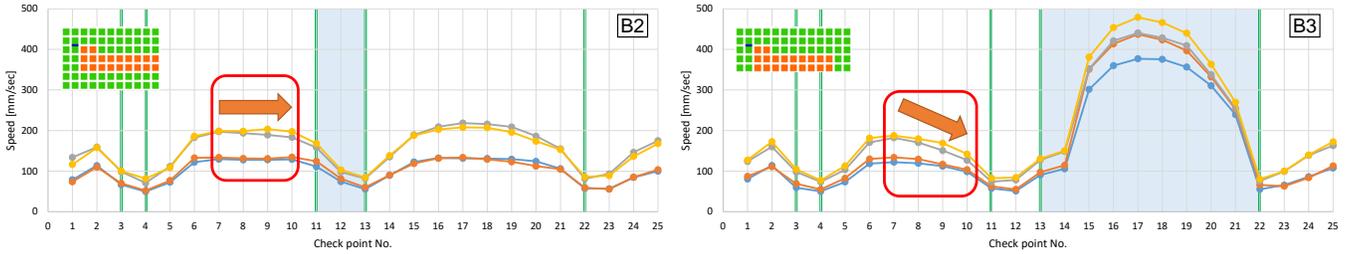


Figure 13: Speed profiles in Experiment 2. Green vertical lines indicate corners, and unconstrained areas are shaded in blue.

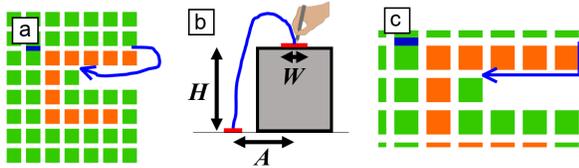


Figure 14: Roundabout movement in layout-C2.

this crossing motion does not involve steering between the top orange icons and bottom green ones (Figure 14c), as the ratio of the path distance to the width is 3, which does not require closed-loop steering [34].

Lastly, we evaluated our proposed model, which sums the difficulties of segmented steering, pointing, and crossing motions (last row in Table 2). The pointing and crossing terms significantly improve model fitness, and it is thus possible to accurately estimate the time for an entire lasso motion.

Discussion of Experiment 2

Model Fitting. In contrast to Experiment 1, the segmented version of the global steering model did not improved the model fitness much over the baseline model. Interestingly, adding terms for either pointing or crossing motions showed significant improvements, which expands on Experiment 1 where only pointing motions helped. Finally, our new model showed a significantly better fit, according to the *AIC* [5]. The *SDs* of the deviations ($MT_{\text{predicted}} - MT_{\text{observed}}$) for the rows of Table 2, respectively. Similar to Experiment 1, our model again predicts *MT* more accurately. Thus, our new model enables designers to predict how long a given lasso motion may take, regardless if the targets are fully or partially constrained by other objects.

Speed Profile and Cursor Trajectory. As expected, participants made curved “roundabout” strokes in unconstrained areas (Figure 11). These sloppy movements allowed participants to move the stylus more quickly, and speed increased more in long unconstrained areas. This is visible in the areas highlighted in blue in Figure 13. In addition, we identified another property of unconstrained areas; as annotated by

the red rectangle for *layout-B2* in Figure 13, participants did not have to decelerate in advance of the unconstrained area. Thus, they could keep the speed high until exiting the third constrained segment. In comparison, in the fully-constrained condition (*layout-B1*), participants had to decelerate in advance of the third corner, and hence the speed did not reach the maximum possible for the given path width.

In interviews after the experiment, 7 out of 12 participants stated that they were able to accelerate in unconstrained areas. One of them emphasized that they experienced this even when the unconstrained area was only for a single icon (the second unconstrained area in *layout-E2*). In contrast, 2 out of 12 participants identified a disadvantage of unconstrained areas: when coming from an unconstrained area and entered a constrained segment, they could not decrease the speed and accidentally hit an icon, because they did not control the speed well enough for the constrained path width W .

5 GENERALIZATIONS OF THE MODEL

In this work we focused mainly on model fitting for different icon layouts that include constrained and unconstrained areas. Our results are somewhat limited by the experimental conditions, e.g., we tested only two values of margin W and icon size S for grid arrangements. Yet, we expect that our results generalize to a larger range of characteristics, as long as the steering, pointing, and crossing laws hold. Also, in both experiments, we tested only clockwise strokes performed by right-handed participants, to control any potentially negative effect of hand occlusion. Moreover, as users naturally slow down if they cannot see the path ahead, we also expect that users naturally avoid occlusions to make their task easier.

Nancel and Land carefully avoided hand occlusions when developing a single curved path for steering tasks [28]. Yet, a lassoing task requires a loop. Thus, this problem cannot be completely ignored. To increase internal validity, we fixed the start position at the top-left of the orange icons. Yet, in actual lassoing tasks users can determine where to start lassoing, which could affect performance. Also, automatically closing the stroke makes lassoing easier. We believe that our work directly generalizes to such functionality, as it is trivial to exclude the last term for “closing the loop”. Still, auto-closing

Table 2: Model fitting results with 95% CIs [lower, upper] for Experiment 2. #s, #p, and #c indicate the numbers of steering, pointing, and crossing motions, respectively.

Model	Formulation	a_s	b_s	a_p	b_p	a_c	b_c	adj. R^2	AIC
Steering (global)	$MT = a_s + b_s(ID_s)$	2522 [1524, 3520]	77.0 [61.7, 92.3]					0.755	577
Steering (segmented)	$MT = a_s(\#s) + b_s(ID_s)$	326 [219, 434]	79.5 [67.5, 91.5]					0.795	569
Steering (segmented) with pointing	$MT = a_s(\#s) + b_s(ID_s) + a_p(\#p) + b_p(ID_p)$	943 [767, 1118]	54.0 [42.7, 65.3]	-1560 [-2001, -1120]	341 [216, 466]			0.930	534
Steering (segmented) with crossing	$MT = a_s(\#s) + b_s(ID_s) + a_c(\#c) + b_c(ID_c)$	273 [209, 337]	78.4 [71.7, 85.2]			-795 [-1229, -361]	413 [287, 540]	0.940	529
Proposed	$MT = a_s(\#s) + b_s(ID_s) + a_p(\#p) + b_p(ID_p) + a_c(\#c) + b_c(ID_c)$	-102 [-817, 613]	68.8 [57.9, 79.8]	-95.6 [-1004, 813]	201 [103, 298]	-139 [-766, 488]	401 [271, 531]	0.971	506

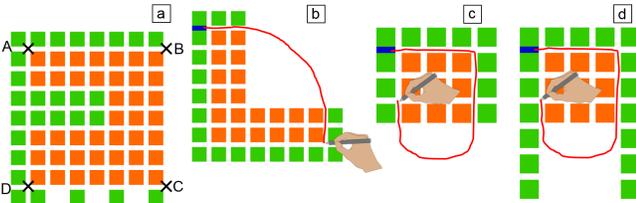


Figure 15: Lassoing situations not investigated in this work. See text for details.

introduces strategic issues. For example, in Figure 15a, the distances A-B and B-C are the same, but passing through A-B is slower. Therefore it seems that auto-closing the loop at A-B is better. However, the most effective path to be auto-connected is likely C-D, because successive obstacles further decrease the steering speed compared to a fully-constrained area [42]. A topic of further research is thus to identify how users determine the lasso start and end positions.

Other untested icon layouts which possibly affect the MT are shown in Figure 15b–d. In (b), and compared with the fully-constrained condition (*layout-A* used in Experiment 1), the unconstrained area allows a “cut-off” motion. Although we did not include such a layout in our experiments, we believe that the MT for this unconstrained area can be accurately modeled as a crossing motion with A of its diagonal distance. Comparing (c) and (d), the green icons at the bottom left in (d) objectively should not affect user performance (unless users intentionally overshoot). One could argue that users might perceive a different task difficulty and change their behavior. Yet, as we observed very little overshooting, see the movement trails in Figure 11, we believe this to be a minor issue that does not affect our model substantially.

Limitations and Future Work

As that we tested only two values of W and S , and we investigated only a single lasso criterion (Figure 3d), our study is limited by the experimental conditions. Another limitation

is that we explored only lassoing of icons in regular and orthogonally arranged grids; thus we cannot claim that our current model can be applied to all lassoing tasks. While such grids are commonplace in file and photo management applications, other applications involve situations with irregular sized, shaped, and positioned objects.

Still, we assume that our lasso path segmentation approach might generalize to such tasks by integrating additional factors, e.g., path curvature [27, 28], width-changing paths [39, 40], and successive path segments [41, 42]. As our above work shows that a summation of task-dependent terms corresponding to each path segment is a good approach to predict movement time, we expect this approach to generalize to such, more complex, tasks as well. To achieve our long-term goal of modeling general lasso tasks, we plan to explore such extensions in future work through a series of experiments that enables us to verify the utility of each individual factor.

6 CONCLUSION

We derived models to predict movement times for lassoing tasks under both fully- and partially-constrained conditions. Results showed that summing indices of difficulty for steering, pointing (stopping at a corner), and crossing (entering a constrained area from unconstrained area) movements can accurately capture user performance and yields significantly better results than other approaches. The margin between icons, icon size, and icon layout (naturally) significantly affected movement time. Beyond that, we showed that careful segmentation of the path into different tasks and summing corresponding model terms accurately estimates a whole lasso motion. We also discussed how our work can be generalized to several conditions we did not experimentally evaluate here. In the future, we plan to further generalize our work towards lassoing with more irregular objects and arrangements.

REFERENCES

- [1] Johnny Accot and Shumin Zhai. 1997. Beyond Fitts' law: models for trajectory-based HCI tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '97)*. 295–302. <https://doi.org/10.1145/258549.258760>
- [2] Johnny Accot and Shumin Zhai. 1999. Performance Evaluation of Input Devices in Trajectory-based Tasks: An Application of the Steering Law. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. ACM, New York, NY, USA, 466–472. <https://doi.org/10.1145/302979.303133>
- [3] Johnny Accot and Shumin Zhai. 2001. Scale effects in steering law tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '01)*. 1–8. <https://doi.org/10.1145/365024.365027>
- [4] Johnny Accot and Shumin Zhai. 2002. More than dotting the i's – foundations for crossing-based interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '02)*. 73–80. <https://doi.org/10.1145/503376.50339>
- [5] Hirotugu Akaike. 1974. A new look at the statistical model identification. *IEEE Trans. Automat. Control* 19, 6 (Dec 1974), 716–723. <https://doi.org/10.1109/TAC.1974.1100705>
- [6] Georg Apitz, François Guimbretière, and Shumin Zhai. 2008. Foundations for Designing and Evaluating User Interfaces Based on the Crossing Paradigm. *ACM Trans. Comput.-Hum. Interact.* 17, 2, Article 9 (May 2008), 42 pages. <https://doi.org/10.1145/1746259.1746263>
- [7] Per Bjerre, Allan Christensen, Andreas Køllund Pedersen, Simon André Pedersen, Wolfgang Stuerzlinger, and Rasmus Stenholt. 2017. Predictive Model for Group Selection Performance on Touch Devices. In *Human-Computer Interaction. Interaction Contexts - 19th International Conference, HCI International 2017, Vancouver, BC, Canada, July 9-14, 2017, Proceedings, Part II*. 142–161. https://doi.org/10.1007/978-3-319-58077-7_12
- [8] Hoda Dehmeshki and Wolfgang Stuerzlinger. 2009. ICE-Lasso: An enhanced form of Lasso selection. In *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*. 630–635. <https://doi.org/10.1109/TIC-STH.2009.5444424>
- [9] Hoda Dehmeshki and Wolfgang Stuerzlinger. 2006. Using perceptual grouping for object group selection. In *Extended Abstracts Proceedings of the 2006 Conference on Human Factors in Computing Systems, CHI 2006, Montréal, Québec, Canada, April 22-27, 2006*. 700–705. <https://doi.org/10.1145/1125451.1125593>
- [10] Jack Tigh Dennerlein, David B. Martin, and Christopher Hasser. 2000. Force-feedback Improves Performance for Steering and Combined Steering-targeting Tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '00)*. ACM, New York, NY, USA, 423–429. <https://doi.org/10.1145/332040.332469>
- [11] Colin G. Drury. 1971. Movements with lateral constraint. *Ergonomics* 14, 2 (1971), 293–305. <https://doi.org/10.1080/00140137108931246>
- [12] 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>
- [13] Tamar Flash and Neville Hogan. 1985. The coordination of arm movements: an experimentally confirmed mathematical model. *The Journal of Neuroscience* 5, 7 (1985), 1688–1703.
- [14] Khai-Chung Gan and Errol R. Hoffmann. 1988. Geometrical conditions for ballistic and visually controlled movements. *Ergonomics* 31, 5 (1988), 829–839. <https://doi.org/10.1080/00140138808966724>
- [15] 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 (CHI '05)*. ACM, New York, NY, USA, 281–290. <https://doi.org/10.1145/1054972.1055012>
- [16] Yves Guiard, Renaud Blanch, and Michel Beaudouin-Lafon. 2004. Object Pointing: A Complement to Bitmap Pointing in GUIs. In *Proceedings of Graphics Interface 2004 (GI '04)*. Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 9–16. <http://dl.acm.org/citation.cfm?id=1006058.1006060>
- [17] Errol R. Hoffmann. 2009. Review of models for restricted-path movements. *International Journal of Industrial Ergonomics* 39, 4 (2009), 578–589. <https://doi.org/10.1016/j.ergon.2008.02.007> Special issue: Felicitating Colin G. Drury.
- [18] Errol R. Hoffmann and Calvin Ngo. 2017. Effects of path angle changes on the times for ballistic and visually-controlled two-component arm movements. *International Journal of Industrial Ergonomics* 61 (2017), 13–21. <https://doi.org/10.1016/j.ergon.2017.05.001>
- [19] Errol R. Hoffmann and Ilyas H. Sheikh. 2012. Goal-directed aimed movements with path obstructions. *Ergonomics* 55, 8 (2012), 946–962. <https://doi.org/10.1080/00140139.2012.681809>
- [20] Seung-Kweon Hong and Seungwan Ryu. 2007. Human Performance Model for Combined Steering-Targeting Tasks. In *Engineering Psychology and Cognitive Ergonomics*, Don Harris (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 306–315. https://doi.org/10.1007/978-3-540-73331-7_33
- [21] Sergey Kulikov and Wolfgang Stuerzlinger. 2006. Targeted Steering Motions. In *CHI '06 Extended Abstracts on Human Factors in Computing Systems (CHI EA '06)*. ACM, New York, NY, USA, 983–988. <https://doi.org/10.1145/1125451.1125640>
- [22] Gordon Kurtenbach and William Buxton. 1993. The Limits of Expert Performance Using Hierarchic Marking Menus. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems (CHI '93)*. ACM, New York, NY, USA, 482–487. <https://doi.org/10.1145/169059.169426>
- [23] Edward Lank and Eric Saund. 2005. Sloppy Selection: Providing an Accurate Interpretation of Imprecise Selection Gestures. *Comput. Graph.* 29, 4 (Aug. 2005), 490–500. <https://doi.org/10.1016/j.cag.2005.05.003>
- [24] 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
- [25] Sachi Mizobuchi and Michiaki Yasumura. 2004. Tapping vs. Circling Selections on Pen-based Devices: Evidence for Different Performance-shaping Factors. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 607–614. <https://doi.org/10.1145/985692.985769>
- [26] M. Ali Montazer and Colin G. Drury. 1989. A test of the Beggs' model for self-paced movements. *Ergonomics* 32, 5 (1989), 497–511. <https://doi.org/10.1080/00140138908966120>
- [27] M. A. Montazer, C. G. Drury, and M. H. Karwan. 1988. An optimization model for self-paced tracking on circular courses. *IEEE Transactions on Systems, Man, and Cybernetics* 18, 6 (Nov 1988), 908–916. <https://doi.org/10.1109/21.23090>
- [28] Mathieu Nancel and Edward Lank. 2017. Modeling User Performance on Curved Constrained Paths. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 244–254. <https://doi.org/10.1145/3025453.3025951>
- [29] Robert Pastel. 2006. Measuring the Difficulty of Steering Through Corners. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '06)*. ACM, New York, NY, USA, 1087–1096. <https://doi.org/10.1145/1124772.1124934>
- [30] Nicolas Rashevsky. 1959. Mathematical biophysics of automobile driving. *Bulletin of Mathematical Biophysics* 21, 4 (1959), 375–385. <https://doi.org/10.1007/BF02477896>

- [31] Ransalu Senanayake, Errol R. Hoffmann, and Ravindra S. Goonetilleke. 2013. A model for combined targeting and tracking tasks in computer applications. *Experimental Brain Research* 231, 3 (01 Nov 2013), 367–379. <https://doi.org/10.1007/s00221-013-3700-4>
- [32] 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* 61, 6 (2004), 751–789. <https://doi.org/10.1016/j.ijhcs.2004.09.001>
- [33] Sven Strothoff, Wolfgang Stuerzlinger, and Klaus Hinrichs. 2015. Pins 'n' Touches: An Interface for Tagging and Editing Complex Groups. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces (ITS)*. ACM, New York, NY, USA, 191–200. <https://doi.org/10.1145/2817721.2817731>
- [34] Namal Thibbotuwawa, Ravindra S. Goonetilleke, and Errol R. Hoffmann. 2012. Constrained Path Tracking at Varying Angles in a Mouse Tracking Task. *Human Factors* 54, 1 (2012), 138–150. <https://doi.org/10.1177/0018720811424743>
- [35] Namal Thibbotuwawa, Errol R. Hoffmann, and Ravindra S. Goonetilleke. 2012. Open-loop and feedback-controlled mouse cursor movements in linear paths. *Ergonomics* 55, 4 (2012), 476–488. <https://doi.org/10.1080/00140139.2011.644587>
- [36] Paolo Viviani and Tamar Flash. 1995. Minimum-jerk, two-thirds power law, and isochrony: converging approaches to movement planning. *Journal of Experimental Psychology: Human Perception and Performance* 21, 1 (1995), 32–53. <https://doi.org/10.1037/0096-1523.21.1.32>
- [37] Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. 2007. Gestures Without Libraries, Toolkits or Training: A \$1 Recognizer for User Interface Prototypes. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology (UIST '07)*. ACM, New York, NY, USA, 159–168. <https://doi.org/10.1145/1294211.1294238>
- [38] Pengfei Xu, Hongbo Fu, Oscar Kin-Chung Au, and Chiew-Lan Tai. 2012. Lazy Selection: A Scribble-based Tool for Smart Shape Elements Selection. *ACM Trans. Graph.* 31, 6, Article 142 (Nov. 2012), 9 pages. <https://doi.org/10.1145/2366145.2366161>
- [39] Shota Yamanaka and Homei Miyashita. 2016. Modeling the Steering Time Difference Between Narrowing and Widening Tunnels. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 1846–1856. <https://doi.org/10.1145/2858036.2858037>
- [40] Shota Yamanaka and Homei Miyashita. 2016. Scale Effects in the Steering Time Difference Between Narrowing and Widening Linear Tunnels. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction (NordiCHI '16)*. ACM, New York, NY, USA, Article 12, 10 pages. <https://doi.org/10.1145/2971485.2971486>
- [41] Shota Yamanaka, Wolfgang Stuerzlinger, and Homei Miyashita. 2017. Steering Through Sequential Linear Path Segments. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 232–243. <https://doi.org/10.1145/3025453.3025836>
- [42] Shota Yamanaka, Wolfgang Stuerzlinger, and Homei Miyashita. 2018. Steering Through Successive Objects. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 603, 13 pages. <https://doi.org/10.1145/3173574.3174177>
- [43] Shumin Zhai, Jing Kong, and Xiangshi Ren. 2004. Speed-accuracy tradeoff in Fitts' law tasks: on the equivalency of actual and nominal pointing precision. *International Journal of Human-Computer Studies* 61, 6 (2004), 823–856. <https://doi.org/10.1016/j.ijhcs.2004.09.007>
- [44] Shumin Zhai and Per-Ola Kristensson. 2003. Shorthand Writing on Stylus Keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, NY, USA, 97–104. <https://doi.org/10.1145/642611.642630>