Deriving Selection Techniques for GUIs based on the Multiple Process Model

Difeng Yu
Department of Computer Science
University of Copenhagen
Copenhagen, Denmark
diyu@di.ku.dk

James Roberts
Liverpool John Moores University
Liverpool, United Kingdom
j.w.roberts@ljmu.ac.uk

Kasper Hornbæk
Department of Computer Science
University of Copenhagen
Copenhagen, Denmark
kash@di.ku.dk

Joanna Bergström
Department of Computer Science
University of Copenhagen
Copenhagen, Denmark
joanna@di.ku.dk

Abstract

Designing efficient selection techniques for graphical user interfaces (GUIs) is fundamental in HCI research. We derive selection techniques based on the multiple process model, a theory that details the motor control processes during goal-directed movements. Specifically, we deduce three theoretical assumptions on how control processes of pre-planning, impulse control, and limb-target control could influence selection movements when adjusting GUI elements, including visual feedback, cursor position, and target position. Corresponding to our assumptions, we develop three techniques that hide the cursor when a target is highlighted, snap the cursor when selection begins, and expand clustered objects during selection movements. After that, we pre-register the assumptions and research methodology and evaluate the techniques in three crowdsourcing-based pointing studies. Our results show that all techniques improved the selection efficiency compared to established baselines. We further discuss the design implications and reflect on how we derived techniques from theory.

CCS Concepts

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

Keywords

Cursor, input, object selection, pointing, target selection

ACM Reference Format:
Difeng Yu, James Roberts, Kasper Hornbæk, and Joanna Bergström. 2025. Deriving Selection Techniques for GUIs based on the Multiple Process Model. In *CHI Conference on Human Factors in Computing Systems (CHI '25), April 26–May 01, 2025, Yokohama, Japan.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3706598.3713089

CHI '25, Yokohama, Japan

ACM ISBN 979-8-4007-1394-1/25/04
https://doi.org/10.1145/3706598.3713089

1 Introduction

Target selection is a fundamental task in graphical user interfaces (GUIs) such as those in desktops, mobile devices, and mixed reality systems. The targets include menus, icons, buttons, and 3D objects, which can vary in their sizes and inter-distances, and can be arranged in grid, circular, randomized, and other layouts [2, 5]. It is thus crucial to develop efficient techniques for various selection tasks. To develop such techniques, the field of human-computer interaction (HCI) has accumulated many principles and guidelines. For instance, one key principle to improve selection performance is to reduce the effective distance from the cursor to the target (D_e) and enlarge the target's effective size (W_e) [5]. This has been achieved by dynamically adjusting the cursor [34], target [50], and control-display gain [12].

The field of motor control focuses on understanding how goal-directed reaching movements, such as target selection in GUIs, are planned and executed [25, 26, 52]. It has built models to describe these movements, which have provided important theoretical background for design rationales in HCI. For example, the *optimized submovement model* [51] proposed in 1988 has been used to justify the designs that decrease D_e to expedite the fast *ballistic* movement and increase W_e to facilitate the feedback-based *correction* movement [5, 17].

The *multiple process model* [23, 26] is a theory about the underlying control processes—specifically *pre-planning*, *impulse control*, and *limb-target control*—that give rise to the ballistic and correction components described in previous models. The model quantifies when and to what extent these processes occur, which has been supported by extensive empirical evidence. By providing detailed explanations of the underlying mechanics, the model enables insights into how dynamic changes in GUI elements can affect selection movements and, consequently, user performance.

In this research, we derive new selection techniques for GUIs based on the multiple process model. We first propose three theoretical assumptions about how control processes may influence selection movements and user performance due to the manipulation of GUI elements, including hiding the cursor when a target is highlighted, snapping the cursor when selection begins, and expanding clustered objects during selection movements. Corresponding to each assumption, we develop three techniques (*Cloaking*, *Pulsing*, and *Unfurling*) that take advantage of these control processes. After that, we pre-register the assumptions and research methodology and conduct three crowdsourcing-based pointing studies to evaluate the techniques. The results show that all techniques improved the selection efficiency compared to established baselines. Based on the study results, we discuss design implications. Finally, we reflect on how to derive techniques from theory through what we term *justified concepts*.

2 Related Work

We first introduce key concepts in HCI that are relevant to our work in designing target selection techniques for GUIs. We then highlight several goal-directed reaching models that have provided theoretical background for the selection techniques. We finally discuss the multiple process model that serves as a base theory to derive our techniques.

2.1 Target Selection in GUIs

In GUIs, target selection usually involves users controlling an end effector, such as a cursor, virtual hand, or pointer, to both *indicate* (e.g., by pointing) and *confirm* (e.g., by pressing a button) their selection of a target. The field of HCI has proposed many guidelines and techniques to enhance selection performance and experience [2, 5].

One key design guideline is to convey what the system knows about the user input through *semantic feedback*. For example, GUIs often display where the user is pointing (e.g., by presenting a visual cursor) and which object the user is targeting (e.g., by visually highlighting the boundary of an object). Such feedback is typically an integral part of GUIs in modern personal computers, helping to reduce user errors and improve efficiency [1].

Another rationale for improving user performance is to reduce movement distance D_e and increase target size W_e. Many *selection facilitation techniques* follow this rationale by altering the positions and sizes of the cursor and candidate objects [5, 34]. For example, to reduce D_e, snapping techniques [7, 36] eliminate the empty space between the cursor and the target by making the cursor jump towards the target. To increase W_e, target expansion techniques [37, 50] enlarge the size of the target. To reduce D_e and increase W_e simultaneously, techniques such as *pointer acceleration* [17] and *semantic pointing* [12] dynamically adjust the cursor velocity or the control-display gain based on the relative positions between the cursor and objects.

To enable more precise selection of clustered or occluded targets, a selection technique may introduce additional steps to minimize user errors. For example, visual menu techniques [4] and other *progressive refinement techniques* [3, 29, 84] reorganize densely clustered targets into more accessible layouts, such as decision trees, grids, or circular arrangements, to facilitate easier selection and disambiguation. Other techniques may incorporate multiple modalities (e.g., eye gaze for faster pointing and hand for more precise refinements [46]) or movement trajectory information (e.g., smooth pursuit [77] or trajectory-based target prediction methods [18, 41]).

The field of object selection in GUIs continues to evolve and expand with new techniques, especially with the challenges arising from advancements in technologies such as mobile devices, smartwatches, and mixed reality systems, which provide unique challenges, including limited screen sizes and the complexities of 3D interactions [82]. Many techniques could find their theoretical root in goal-directed reaching models.

2.2 Goal-Directed Reaching Models

Target selection in GUIs can be viewed as a goal-directed reaching task, which has been studied and modeled within the field of motor control. In 1899, Woodworth identified two main components in such movements: an initial adjustment determined by the first movement impulse followed by finer adjustments when approaching closer to the target [79]. Over more than a century of development, researchers have proposed many models [25, 52, 64] to describe these movements and explain Fitts's law [31], such as the *iterative correction model* [21, 45], the *impulse variability model* [70], and the *single correction model* [9, 10].

The *optimized submovement model* [51, 52, 64] proposed by Meyer et al. in 1988 is perhaps the most familiar one to the HCI audience [55]. According to the model, a rapid movement impulse, often dubbed as the *ballistic movement*, is first initiated to hit the target. However, the movement's spatial accuracy is imperfect, and its endpoint may fall outside the target. This occurs because noise increases with the magnitude of force, which rises in more ballistic movements—a phenomenon known as the speed-accuracy trade-off [70]. In conditions where the endpoint falls outside of the target, one or more feedback-based *correction movements* are employed to remedy the error until the target is reached. The model can be used to justify the design of many GUI selection techniques [5, 17]. For example, techniques have decreased D_e in the rapid movement impulse and increased W_e during the correction phase to enhance user performance [5]. They have also used fast modalities (e.g., gaze) to cover for D and more precise modalities (e.g., hand) to fine-tune for W, because the later stages of the movement is supervised by the visual feedback [46].

Although the models describe the formulation of goal-directed movement, they do not explain the underlying control processes that lead to this specific formulation. As a result, it is challenging to derive new designs from these models, as the mechanisms by which the processes might influence or alter a movement remain unclear [20]. The *multiple process model* represents a recent attempt to characterize the control processes involved before and during movement.

2.3 The Multiple Process Model

The multiple process model of goal-directed reaching was introduced by Elliott et al. in 2010 [23] and updated in 2017 [26]. Based on extensive experimentation, the authors realized that while there are two main, identifiable components in most reaching movements (i.e., the ballistic and correction parts), multiple processes could give rise to these components. The key processes are: pre-planning, impulse control, and limb-target control.

Pre-planning occurs before initiating a goal-directed movement to encode relevant information, like perceived target size and distance, which will later influence the generation of motor commands, such as the required force. The planning process aims to aid the optimal decision of movement speed, accuracy, and also importantly, energy expenditure [24, 26]. For example, motor control research reveals that users typically undershoot a target with their primary movements because overshooting is more costly to correct [25, 27]. Notably, the planning and optimization process also considers prior knowledge of, for example, error feedback from previous trials [80] and the availability of online feedback throughout the movement [26, 39] (e.g., the presence of a cursor and the visual indication of hovering on a target in GUI). Therefore, when designing a GUI technique, it can be crucial to determine how such feedback may influence users' movement strategies.

Impulse control and *limb-target control* are two processes that govern an ongoing goal-direct reaching movement. The impulse control process compares the actual velocity and direction of a limb (or an end effector such as a cursor) to a user's expectations about the limb trajectory [26, 71]. The limb-target control process exerts discrete error reduction by assessing the relative positions of the limb and the target [26, 61, 67, 71]. While impulse control begins almost immediately after movement initiation, limb-target control relies on feedback-based processes that unfold later during the movement. The processes suggest that users will likely adjust their movement dynamics when the end effector or the target context changes.

To create interfaces with better usability [44], many GUI selection techniques modify such (visual) information constantly. For instance, selection techniques may dynamically adjust the cursor movement speed [12, 47, 60] or snap/warp the cursor position [85] to enable faster or more precise selection, essentially influencing the impulse control process. They may also enlarge the size of a target [50] or the cursor [5, 34] or expand a cluster of objects [29, 35, 84] during cursor movements, which could be relevant to limb-target control process. Consequently, it can be essential to consider how such changes influence user movements, which are governed by impulse and limb-target control, and how to leverage these control processes.

3 Research Overview

This research aims to derive selection techniques in GUIs based on the multiple process model [23, 26]. We deduced how the control processes, including pre-planning, impulse control, and limb-target control, may influence the user's movements and performance if we modify specific GUI elements, including visual feedback, cursor position, and target position. Based on these deductions, we derived three selection techniques that take advantage of these control processes to improve object selection on GUIs. We then pre-registered our assumptions, techniques, and evaluation methodology (i.e., task scenarios, metrics, and analysis procedure) to prepare for hypothesis testing. After that, we conducted three crowdsourcing-based pointing studies to verify whether our designs could bring meaningful improvements to HCI tasks.

3.1 Assumptions and Techniques

Here, we present an overview of the theoretical assumptions and the techniques to help readers grasp the core ideas. A more detailed explanation will follow in the subsequent sections.

3.1.1 Study 1: invisible cursor and indication feedback. We posit that hiding the cursor when selection indication feedback is enabled (e.g., a target is highlighted) allows faster selection than showing the cursor all the time. The reason is that the former eliminates the lingering between two alternative strategies (triggering a selection based on the indication feedback *vs.* manually placing the cursor to a designated location within the target). *Cloaking* hides the cursor based on this assumption.

3.1.2 Study 2: cursor snapping. We posit that introducing a cursor snap at the beginning of a selection movement can seamlessly shorten the aiming distance and, therefore, the selection time. The reason is that the impulse control process can quickly detect the mismatches between the expected and actual cursor velocity and correct the positional and directional deviations caused by the snap. *Pulsing* is developed and fine-tuned based on this assumption.

3.1.3 Study 3: clustered object selection. We posit that expanding an object cluster during a selection movement allows users to redirect their ongoing movement towards the intended target, which can reduce their selection time and cursor movement distance. The reason is that the limb-target control process can reduce the positional difference between the cursor and the target on the fly. *Unfurling* is derived from this rationale for selecting clustered objects.

3.2 Methodology

After developing the hypotheses and designing the techniques, we formulated the task scenarios and analysis procedure and pre-registered them on the Open Science Framework (OSF) to ensure transparency and reproducibility. The pre-registration is publicly accessible at https://doi.org/10.17605/OSF.IO/NW2GK. The techniques were tested in pointing tasks through crowdsourcing.

3.2.1 Testbed. The testbed is a slightly modified version of Fitts's Ring [73], as demonstrated in Figure 1. The targets are distributed in a circular manner. The cursor starts inside a red circle, and a target appears when the circle turns green. This is conditioned on the cursor staying within a small starting circle for 0.5 seconds to ensure the selection starts with minimal movement. The starting positions follow a clockwise order ($N = 0, 1, ..., 8$), and the target is located at the opposite side of the ring, with the number $T = (N + 5)$ *MOD* 9. Once the target is selected, a trial is completed, and the cursor needs to move to the next starting position. Importantly, a correct selection of the target is required to proceed to the subsequent trial; this controls selection errors.

The original Fitts's Ring [73] assumes the next starting position is the current target position. This does not work for our studies because the effective distances and widths are significantly shifted due to the selection techniques applied, thus the "ideal" starting position of the cursor might not be centered on a target as expected. Therefore, we place the new starting position at the opposite side of the movement direction to minimize its impact on the current selection. In the three user studies, target- and cursor-related parameters were adjusted for different task purposes.

3.2.2 Crowdsourcing study. Participants were recruited from Prolific, a crowdsourcing platform. This recruitment method enables us to obtain a diverse sample with various age groups. Participants

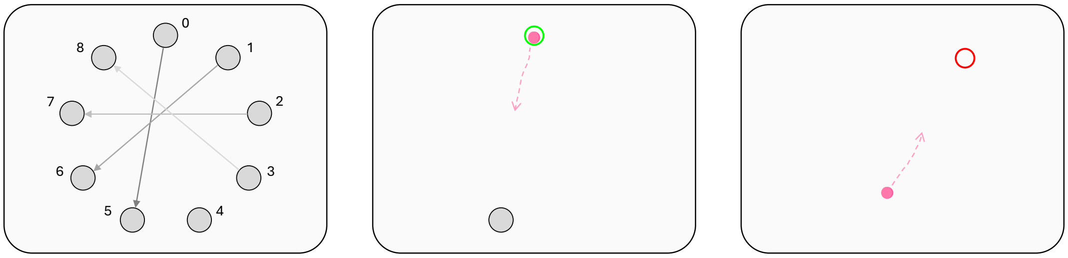

Figure 1: The basic version of the testbed that will be adapted for different task purposes in the three user studies (left). The cursor begins inside a red circle, which turns green with a countdown timer. The objective is to select a circular target (middle). After selecting the target, the cursor needs to travel to a new red circle to start the next trial (right). The cursor's starting positions follow a clockwise order.

may use different mouses (e.g., form factors, DPIs, and transfer functions) and screens (e.g., size and resolution) for completing the task. This diversity is crucial for assessing the real-world applicability and effectiveness of the techniques. Prior to conducting the studies, we had decided to recruit 34 participants for each one of them[1].

3.2.3 Study procedure. Participants first accepted the task. After reading the consent form and completing a demographic questionnaire, they were directed to a WebGL application. There, they calibrated their cursor speed using a slider to a preferred setting, aiming to reach four corners of the screen quickly and precisely. Following calibration, they proceed to the formal experiment, which included practice sessions and formal data collection sessions. After each trial, the performance data and the cursor trajectory data (requested at 100 Hz) were synchronized to an online database[2].

3.2.4 Analysis procedure. We first excluded data of participants who had not completed the study. Next, we discarded outliers where selection times were more than three standard deviations above or below the mean ($mean \pm 3std$) for each participant and condition. We considered these trials as instances where a participant was distracted. We then averaged the trial repetitions per participant per condition. Afterwards, we applied repeated-measures ANOVA (RM-ANOVA) to examine statistically significant effects and Bonferroni-adjusted post-hoc analysis for pairwise comparisons. If the data significantly deviated from a normal distribution, as indicated by Q-Q plots, we transformed the response variable accordingly, for example, through log transformation. When the sphericity assumption was violated, as indicated by Mauchly's test for sphericity, we used the Greenhouse-Geisser correction. Additionally, we calculated effect size measures, including generalized eta-squared (η_G^2) for RM-ANOVA and Cohen's d for comparing two groups. In studies involving speed profile analysis, we used a Savitzky-Golay filter [30, 69] with a 4th-degree polynomial (filter length = 11) to smooth the speed data.

4 Study 1: Invisible Cursor and Indication Feedback

The multiple process model indicates that users are inclined to pre-plan their movements based on available information prior to executing them. The pre-planning does not just aim to minimize time and maximize accuracy but also to optimize energy expenditure [26]. Previous research in motor control has indicated that the pre-planned movement strategy could vary according to differences in, for example, input devices [15], target features (e.g., sizes, distances) [25, 49], user characteristics (e.g., age groups) [78], and visual feedback [22]. One design consideration that is crucial for GUI is how the presentation of *semantic feedback* (i.e., convey what the system knows about the user input [42]) during the selection process may influence pre-planned strategies.

In GUI target selection, two common types of semantic feedback are cursor position [74] and the object under selection indication [2, 53]. A selection cursor is often visible during the movement to illustrate where the user is currently pointing, particularly for relative input devices like a mouse that sense only changes in position (i.e., not absolute positions) [42]. Selection indication feedback, like highlighting the pointed object or its bounding box [53], is also often deployed and thought to be helpful. It conveys which object the system "thinks" a user is about to select, allowing the user to verify and confirm their choice.

However, our theoretical assumptions suggest it might not always be beneficial to show these two types of feedback together, as the presence of the cursor is no longer necessary for selecting the intended target when indication feedback is displayed. The assumptions motivate a technique called *Cloaking*, which hides the selection cursor when selection indication feedback is displayed. *Cloaking* can be especially helpful for more recent predictive systems [41, 57, 81] that aim to show such feedback even earlier than a user has manually moved the cursor onto the target. In the following subsections, we first illustrate why the *Cloaking* technique might work based on its theoretical assumptions. We then present a study to validate the technique's effectiveness and discuss the empirical results.

[1]We used G*Power to estimate the required sample size for our studies. We chose within-subject repeated-measures ANOVA with the following settings: effect size $f = 0.25$ (medium size [19]), α error probability = 0.05, power = 0.8, number of groups = 1 (no between-subject factor), number of measurements = 2 (a new technique vs. a baseline), correlation among repeated measures = 0.5 (default), and nonsphericity correction $\epsilon = 1$ (default). Note that this estimation could be conservative based on our study settings, as an increase in the number of measurements or the correlation among those measures could improve statistical power, thereby reducing the required sample size.
[2]Although the data sampling rate was set to 100 Hz, the actual refresh rate may vary depending on the computer used by each participant.

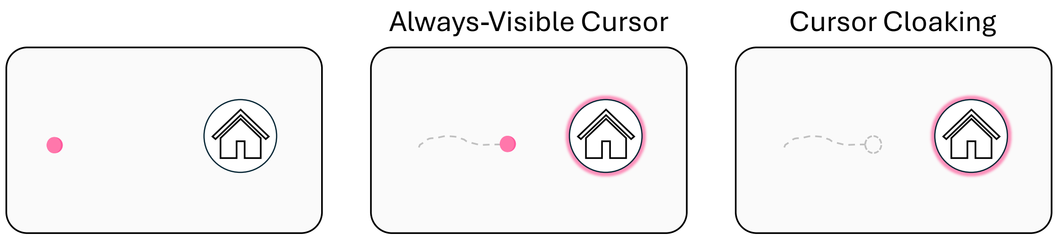

Figure 2: A user plans to move a cursor to select a target icon (left). With an always-visible cursor, users prepare two competing strategies: either trigger the selection when the target is highlighted or manually move the cursor within the target's boundary (middle). *Cloaking* eliminates the second strategy, leaving only the choice to trigger the selection when the target is highlighted (right).

4.1 *Cloaking* and Its Theoretical Assumptions

To select a target in a GUI, users first generate a movement impulse to bring the cursor to the vicinity of the target. They then align the cursor with the target location and press the selection trigger once the target is highlighted (i.e., ready to be selected). The target could be highlighted because the cursor is within the visual boundary of the target or even before the cursor reaches it based on predictions made by selection facilitation mechanisms [6, 81] or when the effective size of the target is larger than its visual size [34, 76, 83].

With a visible cursor, users prepare two competing strategies to ensure optimal task completion. The first strategy is to rely on the manual control of the cursor (i.e., placing the cursor to a designated location within the target boundary). The second strategy is to respond to the selection indication feedback by triggering the selection directly without moving the cursor further. When the target becomes highlighted, the rapid decision of choosing a specific strategy can be considered as a decision diffusion process [32, 63], which requires time to implement and respond. The convergence rate to a decision (i.e., the drift rate) depends on the pre-determined and perceived advantage of one choice over the other (e.g., through learning, users might find one choice is significantly better than the other, so the converging speed will be much faster).

We propose a technique named *Cloaking* (see Figure 2) to eliminate the time-consuming decision process between the two strategies: The cursor is visible during the movement but fades away as soon as the target is highlighted. If the cursor becomes invisible, pressing the trigger as soon as the target is highlighted is always optimal for task completion—this simplifies the selection strategy compared to an always-visible cursor.

Therefore, it is likely that users will be more responsive to selection indication feedback when using *Cloaking*. In other words, the response to such feedback will be faster with *Cloaking*, resulting in a shorter response time than an always-visible cursor. If selection indication feedback is beneficial, like when a selection facilitation technique is applied [41, 81, 83], *Cloaking* should also reduce task completion time compared to an always visible cursor.

4.2 Study Design

The study employs a within-subject design with three independent variables: technique (2 levels), facilitation extent (2 levels), and task difficulty (3 levels). Two dependent variables are used to evaluate user performance: response time and completion time.

4.2.1 Independent variables. The primary aim of the study is to compare *Cloaking* and an always-visible cursor. Additionally, we explore two crucial factors related to task settings—facilitation extent and task difficulty—that might influence the relative performance of the techniques.

- Technique: *Cloaking* and Always-Visible Cursor (*Baseline*).
- Facilitation extent (FE): the extent to which a selection facilitation mechanism is applied relative to the distance between the start and target positions. In this study, we consider two levels of FE, where FE = $\{0, 0.2\}$. FE = 0 means that the target is highlighted and selectable only if the cursor is within the boundary of the visual target. In this case, *Cloaking* is supposed to outperform the baseline if it can eliminate the necessity of fine-tuning the cursor towards a designated location within the target (e.g., the target center). FE = 0.2 means that the target becomes highlighted and selectable when the cursor is within a circular area centered on the target, with a radius equal to 20% of the distance between the start and target positions. That is, a candidate target is predicted and displayed once the cursor has traveled around 80% of the distance—this seems achievable with previous techniques [34] and target prediction models [18, 41, 81]. *Cloaking* should boost the usefulness of facilitation techniques when the target becomes selectable before the cursor reaches it.
- Task difficulty: the combination of target distance D and target size W, which determines how easy or hard it is to complete a selection task. The task difficulty pairs are $\{(8, 0.78), (5.55, 0.37), (8, 0.37)\}$ in this study[3]. If we quantify them as index of difficulty [73] through $ID = \log_2(D/W+1)$, then the $ID \approx \{3.5, 4, 4.5\}$. A higher ID typically means the task is overall more difficult. In our case, decreasing D can diminish the usefulness of the facilitation technique (because of how we calculate FE), and decreasing W reduces the necessity of fine-tuning the cursor position within the target. Both are likely to reduce the performance difference between *Cloaking* and the baseline. We have decided to vary this

[3]Note that the D and W values are presented on a relative scale. Variations in display settings, such as screen and window size, can lead to differences in the absolute number of pixels. Additionally, variations in mouse settings, such as DPI, can influence the actual movement distance of a computer mouse. We account for these differences as user-related factors, reflecting a typical real-world condition. This presentation stays the same for all studies.

Table 1: Results from RM-ANOVA examining the impact of Technique, Technique × Facilitation Extent (FE), and Technique × Task Difficulty on response time and completion time. Significance levels (Sig?) are based on $\alpha = 0.05$.

Dependent Variable	Independent Variable	df_{num}	df_{den}	F	p	η^2_G	Sig?
Response Time	Technique	1	33	4.85	.035	.005	Yes
	Technique × FE	1	33	7.94	.008	.006	Yes
	Technique × Task Difficulty	2	66	0.14	.870	<.001	No
Completion Time	Technique	1	33	2.07	.160	.001	No
	Technique × FE	1	33	14.95	<.001	.005	Yes
	Technique × Task Difficulty	2	66	0.16	.850	<.001	No

factor at three levels to avoid lengthy studies that could cause fatigue and user disengagement.

4.2.2 Dependent variables. We hypothesize that *Cloaking* will outperform the baseline (i.e., an always-visible cursor) with lower response time and completion time. These measurements are thus treated as dependent variables of this study.

- Response time: the time between when the target is highlighted (i.e., indication feedback is enabled) and when the system registers a correct selection (i.e., selection completion).
- Completion time: the time between selection initiation (i.e., the cursor exits the starting circle) and selection completion.

4.2.3 Trial sequence. The study contains 12 experimental conditions (= 2 techniques × 2 FEs × 3 task difficulties), each repeated 9 times (one complete Fitts's Ring). The FEs are nested within the techniques (i.e., Both FEs are tested within one technique before introducing the other). The task difficulties are further nested within the FEs. The orders of all the independent variables are randomized. Before starting each technique and each FE, participants are explicitly informed about the existence of FE (i.e., the highlights appear before or when the cursor is on the target). They practice 9 trials with the easiest D and W combination ($ID = 3$, $D = 5.55$, $W = 0.37$) to get used to the technique and FE.

4.3 Analysis and Results

We collected 3672 trials of data (= 34 participants × 2 techniques × 2 FEs × 2 task difficulties × 9 repetitions). This study included 15 women and 18 men, with a mean age of 29.4 years ($std = 7.6$). No trial was removed based on our data exclusion criteria. The analysis results are summarized in Table 1 and Figure 3. Note that we report only the main effects and interaction effects related to techniques, as our focus is on the effectiveness of the techniques and how it may vary under different experimental conditions. Additionally, the boxplots present transformed rather than raw (absolute) values. More detailed statistical results can be found in the supplementary material (same for the following studies).

4.3.1 Response time. The response time data were transformed with a natural logarithm $\ln(x)$ because response times often follow a log-normal distribution. This transformation accounts for the fact that participants can take varying long time to complete a trial, but there are limits on how fast they can be. Results from RM-ANOVA showed that the techniques significantly influenced response time in general ($p = .035$). An interaction effect of Technique × FE

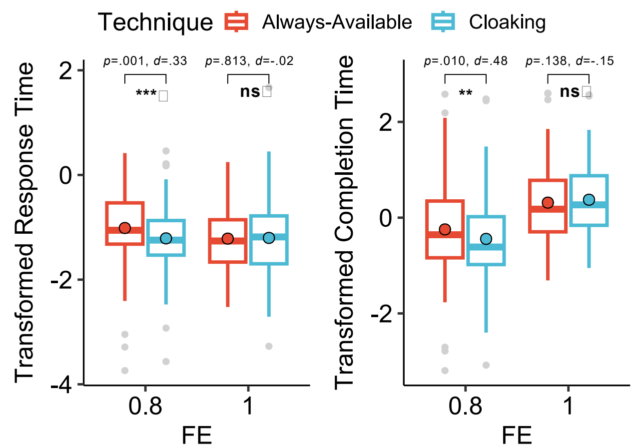

Figure 3: Boxplots display the response time and the completion time across the two levels of Technique and Facilitation Extent (FE). The colored dots with a black outline indicate the mean value for each condition. The response time was transformed with a log transformation, and the completion time was transformed with a Yeo-Johnson transformation ($\lambda = -1.926$). Statistical significance is denoted by p-values derived from pairwise comparisons, with Cohen's d values also reported. The following symbols are used to indicate significance levels: ns (not significant): $p > 0.05$, **: $p \leq 0.01$, and ***: $p \leq 0.001$.

($p = .008$) suggested that the response time of different techniques depends on FE. The post-hoc analysis indicated that the response time of *Cloaking* was significantly lower than an always-available cursor ($p = .001$) when FE = 0.8 (i.e., the selection indication feedback appeared before the cursor reached the target).

4.3.2 Completion time. The completion time data were transformed using a Yeo-Johnson transformation ($\lambda = -1.926$) because the Q-Q plot indicated heavy tails in the data, even after a log transformation. Results from RM-ANOVA suggested an interaction effect of Technique × FE ($p < .001$). Specifically, *Cloaking* led to significantly lower completion time than an always-available cursor ($p = .010$) when FE = 0.8.

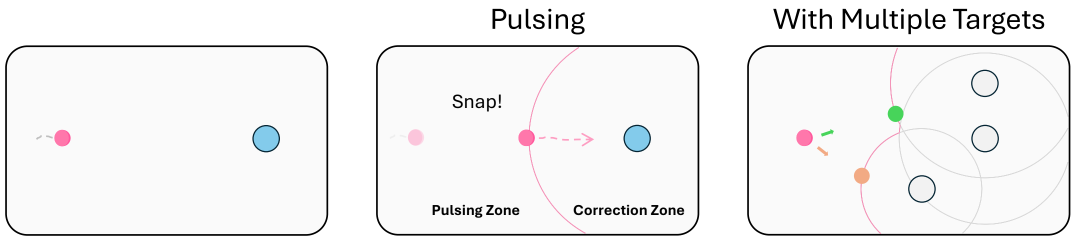

Figure 4: A user moves a cursor to select a target (left). *Pulsing* detects the movement impulse and snaps the cursor to "skip" the early distance that is supposed to be covered by the movement impulse. The user then continues the cursor movement to hit the target as usual (middle). The extent of snapping depends on the distance between the cursor and the target, and the direction of snapping depends on the initial movement direction of the cursor (right).

4.4 Summary

The study suggests that, as compared to an always visible cursor, *Cloaking* can improve users' responsiveness to selection indication feedback and enable faster selection when the feedback is enabled before the cursor reaches the target (e.g., when a selection facilitation mechanism is applied to predict an intended target). There is, however, no apparent benefit of hiding the cursor after the cursor has already entered the visual boundary of the target.

5 Study 2: Cursor Snapping

Impulse control delineates the process of users comparing the actual velocity and direction of the end effector to their internal representations, primarily within the first sub-movement [26]. Such online regulation happens quickly (as little as 85-100ms) and continuously and is considered to be too rapid for a new action plan [26]. The existence of such a process is, for instance, suggested by motor control studies that displaced the background of the cursor and the target. The displacements create an illusion of the cursor moving faster or slower than expected, leading users to extend or shorten the movement impulse [33, 62]. The movement correction/adjustment directed by such a process seems to be rather implicit: research has found that cursor jumps introduced early in a movement can be automatically corrected later without being noticed by the users [66, 68].

Many GUI techniques shift the constant mapping of the input device and the cursor during selection movement to improve user performance. For example, we have mentioned techniques that dynamically increase the cursor gain (i.e., a cursor speed multiplier) [12, 17] to cover the long pointing distance to faraway targets quickly. Cursor snapping techniques such as MAGIC pointing [85] bring the cursor directly to the vicinity of a predicted target (that is indicated by eye gaze) to eliminate the effortful long-distance manual pointing. Other methods snap the cursor to the target when they are close by to allow precise cursor landing [8, 11].

In this study, we discuss a technique called *Pulsing* that snaps the cursor early in a movement to shorten the selection distance. Unlike techniques that increase the cursor gain, *Pulsing* skips the early distance-covering movement entirely and relies on the low-effort impulse control process to "correct" any deviations caused by the snap. Unlike techniques such as MAGIC pointing, *Pulsing* does not rely on additional modalities (e.g., eye gaze) to predict a target region. Through our theoretical assumptions, we determine the extent of snap that allows seamless corrections. In the following, we detail these assumptions and present a study to evaluate *Pulsing*.

5.1 *Pulsing* and Its Theoretical Assumptions

Pulsing is a GUI selection technique that snaps the cursor towards its initial movement direction at the beginning of a selection movement to shorten the travel distance to a target, thereby reducing the selection time (see Figure 4). Because the early cursor movement can be noisy and does not always point directly towards the target [28], we introduce a concept called the correction zone that allows users to "correct" any positional or directional deviations caused by the snap. The correction zone represents an area where every point is within R_{cz} distance. R_{cz} is proportionally defined based on the distance between the initial cursor position and the target. For instance, R_{cz} = 20% means that 20% of the original cursor-target distance is used for movement correction. The rest of the space is called the pulsing zone, and the cursor can only snap to the boundary of the two zones. As a straightforward definition, the pulsing zone for a specific target is $R_{pz} = 1 - R_{cz}$.

We outline two key considerations for determining an appropriate R_{pz}. First, a larger R_{pz} (i.e., a smaller R_{cz}) can save more cursor travel distance if a snap occurs. However, with a larger R_{pz}, *Pulsing* is less likely to trigger a snap because it is more challenging for the noisy initial movement direction to intersect with the zone boundary. Consequently, although a larger R_{pz} could potentially save more distance, the likelihood of a snap occurring also decreases. This suggests an obvious tradeoff when optimizing R_{pz}.

Our second consideration focuses on how users adjust their movement after the cursor snap. The impulse control process quickly begins to correct the mismatch between the expected and perceived cursor velocity [66, 68]. This was achieved by applying an early braking force as the cursor travels faster than expected due to the snap and by regulating the direction of the cursor movement. However, the correction process takes time to complete; otherwise, later in the movement, additional sub-movements that involve greater top-down control may be needed to explicitly correct the endpoint error resulting from the initial impulse [14, 26, 68]. Such conditions are less ideal, as the technique no longer functions seamlessly and could lead to additional movements required to correct errors.

Our conservative estimation is that the maximum of R_{pz} should be 50%. In other words, at least half of the cursor-target distance

Table 2: Results from RM-ANOVA examining the impact of Technique and Technique × Task Difficulty on completion time, cursor total distance, and explicit correction time. Significance levels (Sig?) are based on $\alpha = 0.05$.

Dependent Variable	Independent Variable	df_{num}	df_{den}	F	p	η^2_G	Sig?
Completion Time	Technique	3	99	42.36	<.001	.055	Yes
	Technique × Task Difficulty	6	198	1.26	0.278	.001	No
Cursor Total Distance	Technique	2.33	76.81	12.75	<.001	.110	Yes
	Technique × Task Difficulty	4.41	145.57	3.63	.006	.017	Yes
Explicit Correction Time	Technique	3	99	43.79	<.001	.063	Yes
	Technique × Task Difficulty	6	198	1.31	.254	.001	No

should be used for correction. This is because the process that involves greater top-down control typically happens after reaching the peak velocity [26]. Assuming an ideal movement with a perfectly symmetric speed profile [80], the peak velocity occurs at the 50% distance point. Thus, the deviations caused by the snap should be at least partially corrected before that to enable a seamless transition. When $R_{pz} = 50\%$, a snap can occur if the initial movement direction is within $\arcsin(\frac{1}{2}) = 30^\circ$ from the cursor-to-target vector.

5.2 Study Design

The study uses a within-subject design with two independent variables: technique (4 levels) and task difficulty (3 levels). We examine cursor movement distance and completion time to evaluate the effectiveness of the techniques. We also analyze the explicit correction time to examine whether the cursor snap is seamlessly blended into the whole movement.

5.2.1 Independent variables. The study aims to compare *Pulsing* (with three different snap extents) and regular pointing (i.e., *NP*). We expect the relative performance of the techniques to be significantly influenced by task difficulties (i.e., target distances and sizes).

- Technique: *Pulsing-25*, *Pulsing-50*, *Pulsing-75*, and *NP* (baseline). For the *Pulsing* techniques, the number behind indicates the snap extent (R_{pz}). For instance, *Pulsing-25* means that 75% (=1−25%) of the cursor-target distance is used for correction. In other words, a smaller number indicates a shorter snap distance.
- Task difficulty: target distance D and target size W, which determine the difficulty of a selection task. In this study, the task difficulty pairs are $\{(8, 0.78), (5.55, 0.37), (8, 0.37)\}$, same as the first study. The corresponding *ID*s are $\{3.5, 4, 4.5\}$. This factor influences the amount of time the impulse control process has to "correct" the movement after introducing *Pulsing*. We expect that a longer correction distance will provide more time for adjustments, and a larger target will reduce the need for corrections, leading to a more seamless integration of the snap into the overall movement.

5.2.2 Dependent variables. We hypothesize that *Pulsing-25* and *Pulsing-50* can seamlessly shorten the movement distance, thus reducing the selection time, as compared to *Regular*. *Pulsing-75* may decrease performance compared to the two previous *Pulsing* techniques and may extend the time for explicit correction. Therefore, we examine the performance of the techniques in terms of completion time, cursor total distance, and explicit correction time.

- Completion time: the time from selection initiation to completion.
- Cursor total distance: the total (accumulated) distance traveled by the cursor from selection initiation to its completion.
- Explicit correction time: the time between the end of the primary sub-movement to selection completion. The completion of the movement impulse is determined by a previous parsing algorithm (cf. [51]) that detects (a) velocity zero-crossing from positive + to negative -, (b) acceleration zero-crossing from - to +, and (c) jerk zero-crossing from + to -.

5.2.3 Trial sequence. The study consists of four blocks, each assessing one of the techniques in the three task difficulties. The order of the techniques and task difficulties is randomized, and each task difficulty is repeated 9 times. Before starting each technique, participants practice 9 trials of the technique with $ID = 3$ ($D = 5.55$, $W = 0.37$).

5.3 Analysis and Results

We collected 3669 trials of data (= 34 participants × 4 techniques × 3 task difficulties × 9 repetitions − 3 missing trials). The missing trials were caused by the Internet connection issues to the online database. The study included 14 women and 20 men. The participants' mean age was 31.8 years ($std = 9.0$). No trial was removed based on the data exclusion criteria. The analysis results are summarized in Table 2 and Figure 5.

5.3.1 Completion time. The completion time data were transformed with a Yeo-Johnson transformation ($\lambda = -2.271$). The results from RM-ANOVA showed a significant effect of Technique ($p < .001$). The post-hoc analysis indicated that *NP* and *Pulsing-25* led to the shortest completion time (all $p < .009$), with no statistically significant difference between them ($p = .954$). *Pulsing-50* further led to shorter completion time than *Pulsing-75* ($p < .001$). No interaction effect of Technique × Difficulty was identified ($p = .278$).

5.3.2 Cursor total distance. The cursor total distance data were transformed through a Yeo-Johnson transformation ($\lambda = 1.090$). Results from RM-ANOVA suggested a significant main effect of Technique ($p < .001$). The post-hoc tests showed that *Pulsing-25* and *Pulsing-50* led to the shortest cursor movement distance (all $p < .012$), with a marginally significant difference between them ($p = .082$). *NP* resulted in the longest cursor distance on average, but its difference with *Pulsing-75* was marginal ($p = .081$). There was an interaction effect of Technique × Difficulty ($p = .006$). Most patterns remained consistent across different difficulty levels. Noticeably,

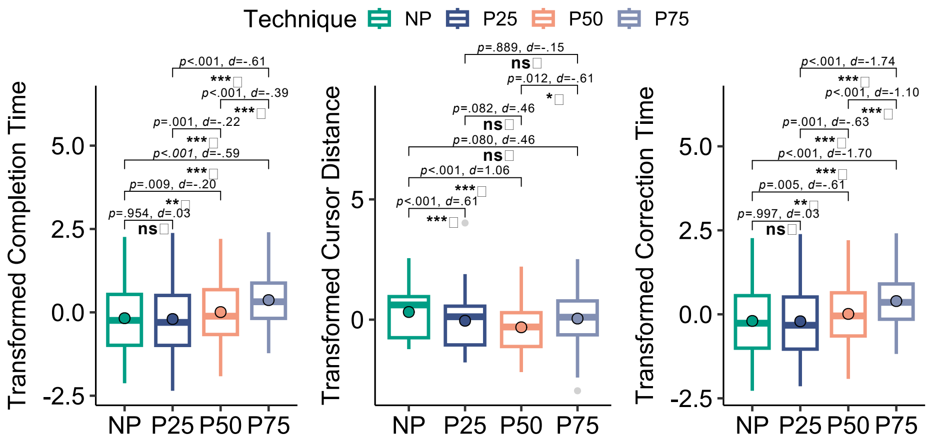

Figure 5: Boxplots display the completion time, the cursor total distance, and the explicit correction time of Technique across Difficulty Level. The colored dots with a black outline indicate the mean value. All data were transformed through Yeo-Johnson transformations (corresponding $\lambda = -2.271$, 1.090, and -2.157). Statistical significance is denoted by p-values derived from pairwise comparisons, with Cohen's d values also reported. The following symbols are used to indicate significance levels: ns: $p > 0.05$, *: $p \leq 0.05$, **: $p \leq 0.01$, and ***: $p \leq 0.001$.

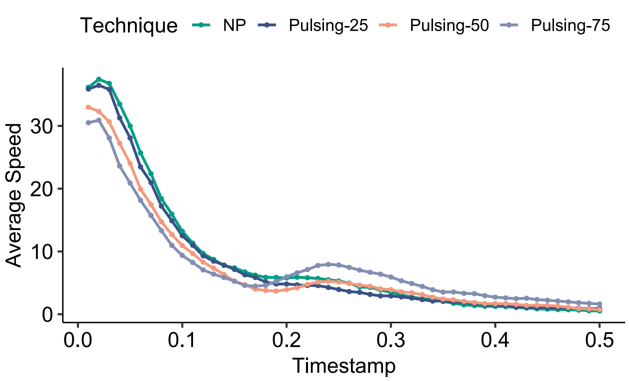

Figure 6: Lineplot of the average cursor movement speed between 0 and 0.5 seconds after exiting the starting circle (also after the snap in the *Pulsing* techniques).

Pulsing-75 did not have statistical difference in cursor total distance compared to *NP* when the target is small and close ($p = .950$), but led to shorter distance in the other two difficulty levels (both $p < .035$).

5.3.3 Explicit correction time. The explicit correction time data were transformed with a Yeo-Johnson transformation ($\lambda = -2.157$). RM-ANOVA and post-hoc tests showed patterns consistent with those observed in completion time.

5.3.4 Speed profile analysis. We examined the speed profiles of the techniques to further investigate the presence of explicit corrections. Figure 6 presents the average cursor movement speed across all trials of a technique between 0 and 0.5 seconds after exiting the starting circle (also after the snap). The results indicated that *Pulsing* led to a slightly lower average speed in the first sub-movement, and a more extended snap resulted in a lower peak velocity. This suggested that a more extended snap may have encouraged participants to pre-plan a more conservative strategy. *Pulsing-50* and *Pulsing-75* led to a noticeable secondary peak after the first sub-movement, which represents explicit corrections.

5.4 Summary

The study suggested that *Pulsing-25* could reduce cursor movement distance compared to regular pointing without requiring additional effort for explicit corrections. The completion time and speed profile of *Pulsing-25* and regular pointing were comparable. This finding indicated that *Pulsing-25* could shorten cursor movement seamlessly. *Pulsing-50* and *Pulsing-75* could also reduce cursor movement distance, but at the cost of more time for explicit correction and longer completion time.

6 Study 3: Clustered Object Selection

Limb-target control minimizes the error vector between the cursor and the target position, sometimes resulting in additional, discrete corrective sub-movements [26]. The process could involve an early "automatic/unconscious" component and a later "voluntary" component [67]. For instance, motor control research that adjusts the target's size or position after movement initiation found that participants started compensating for the perturbation mostly within 180-200 ms (equivalent to visual reaction time), but this compensation could occur as quickly as 100 ms [40, 58, 59, 61].

Target context may also change in GUIs. For example, interruptive notifications may bring new contextual information (e.g., a signal to stop) during selection or reaching movements [65, 75, 81], and target expansion techniques that enlarge the target when the cursor approaches it (like Dock Magnification on macOS) can enable faster target acquisition [50]. More relevant to this study are

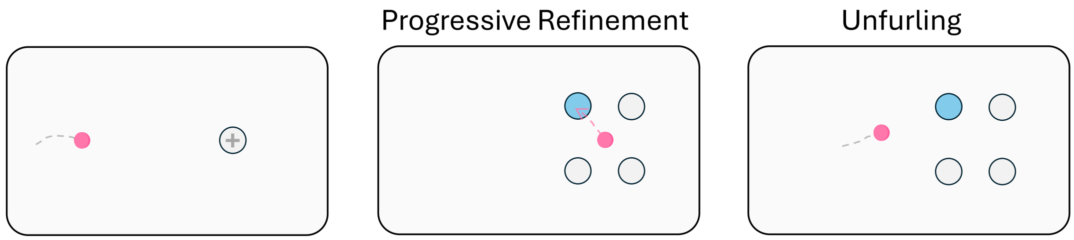

Figure 7: A user moves a cursor towards a target located within a cluster of objects (left). A progressive refinement technique expands the object cluster once the user reaches it, allowing for a more detailed selection of the target (middle). In contrast, *Unfurling* expands the object cluster based on cursor approaching movements to facilitate low-cost redirection towards the target location (right).

methods for selecting a target within a cluster of objects, involving expanding the objects into new locations for disambiguating multiple overlapped objects. The re-arranged objects create a new target context for the later, fine-grained selection of the target (see Figure 7). Such concepts have been applied in, for example, visual menus [4] and other techniques [29, 35, 84] that require two or more steps of selections to acquire the final target (we refer to them as progressive refinement, *PR*, techniques hereafter).

We propose a technique called *Unfurling* for clustered object selection. In contrast to *PR* techniques that often require multiple selection steps (e.g., clicking to expand the target cluster and then reaching the target), *Unfurling* expands potential object clusters during early cursor movements. It leverages the limb-target control process to redirect an ongoing movement towards the target position, without going through a sequential selection and decision process like existing *PR* techniques. We detail our theoretical assumptions in the following subsection. We then report and discuss the results of the user study.

6.1 *Unfurling* and Its Theoretical Assumptions

PR is a powerful family of techniques for precise target selection, especially in cases of overlapping and clustering objects [29, 84]. A typical process begins by reaching a cluster of objects and performing a button click to expand the cluster. This expansion redistributes the objects into distinct regions, narrowing down the intended target for a subsequent selection. In more complicated scenarios, new clusters may appear, and further expansions are performed until the final target is acquired [3]. While such techniques could improve selection precision, the step-by-step process (i.e., first selecting a cluster, then the target) could be further optimized.

The limb-target control process suggests that users aim to minimize the positional differences between the cursor and the target during movement. More specifically, the process can be associated with a "voluntary" visual reaction (180-200 ms) or an even faster "automatic" or "hard-wired" component when associated with positional shifts of the target [38, 67, 72]. While the exact conditions under which a voluntary or an automatic response will be triggered are still being explored, it seems a low-cost solution to displace the target position during an ongoing goal-directed movement (rather than after completing a movement like in *PR* techniques).

Unfurling is motivated by the aforementioned conclusion. Instead of requiring users to complete a selection with multiple steps like *PR* techniques, *Unfurling* expands the relevant object clusters during an ongoing movement (see Figure 7). The "redirection" induced by the limb-target control process theoretically integrates the multiple selection steps into a selection movement with a single target in mind. We infer that *Unfurling* can reduce selection time and cursor movement distances compared to a typical *PR* technique. The null hypothesis is that if the limb-target control process cannot alter a pre-planned selection movement, the performance and movement trajectory of *Unfurling* should be similar to a *PR* technique.

To estimate the relevancy of an object cluster, *Unfurling* needs to be combined with target prediction methods [41, 83, 85] or distance-based thresholding [34]. In this study, we simulate two strategies. The first expands an object cluster immediately after movement initiation (*UStart*), like in cases where we could predict a possible cluster with gaze before hand movement [85]. The second expands a cluster after the velocity peak of a movement impulse (*UPeak*) when the prediction methods become more confident about the intended movement direction [81]. This setting also aligns with a theoretical interest: a "voluntary" reaction typically occurs after the velocity peak, whereas the "automatic" response can occur before it [26, 67].

6.2 Study Design

The study uses a within-subject design with two independent variables: technique (3 levels), cluster density (2 levels), and task difficulty (2 levels). The dependent variables are completion time and total cursor distance.

6.2.1 Independent variables. The goal of the study was to compare *UStart*, *UPeak*, and a *PR* technique. The effectiveness of all the techniques is expected to be influenced by cluster density and task difficulty. Therefore, we systematically vary these parameters to assess the consistency of the observed results.

- Technique: *UStart*, *UPeak*, and a *PR* technique. The *PR* technique requires the cursor to "touch" a cluster to expand it, and then the user clicks on the intended target for selection. For experimental control purposes, the *PR* technique is slightly modified based on the previous literature, which might require multiple button clicks or swipe gestures to select the target [3, 29, 84]. The *PR* technique here can be framed in a similar way as *Unfurling*: object clusters are expanded at the very end of a goal-directed

Table 3: Results from RM-ANOVA examining the impact of Technique, Technique × Cluster Density, and Technique × Task Difficulty on completion time and cursor total distance. Significance levels (Sig?) are based on $\alpha = 0.05$.

Dependent Variable	Independent Variable	df_{num}	df_{den}	F	p	η^2_G	Sig?
Completion Time	Technique	1.55	51.21	77.14	<.001	.117	Yes
	Technique × Cluster Density	2	66	0.07	.930	<.001	No
	Technique × Task Difficulty	2	66	0.26	.770	<.001	No
Cursor Total Distance	Technique	2	66	21.89	<.001	.107	Yes
	Technique × Cluster Density	2	66	0.45	.637	<.001	No
	Technique × Task Difficulty	2	66	0.64	.530	<.001	No

movement, not during it like *UStart* or *UPeak*. In our implementation, *UStart* expands a cluster upon movement initiation (i.e., the cursor exits the starting circle). *UPeak* expands a cluster once the cursor movement distance descends in three consecutive 0.01s intervals.

- Cluster density: the number of objects within a cluster. This study uses cluster density = $\{2, 4\}$. All techniques are likely to decrease their performance as cluster density increases.
- Task difficulty: the combination of distance D and width W that determines the position and size of the cluster and the targets. This study sets the task difficulty pairs as $\{(8, 0.78), (8, 0.37)\}$. The corresponding $ID = \{3.5, 4.5\}$. The distance from the cluster center to an expanded object is fixed as $\sqrt{2}W$. Here, we only vary the target size, as the limb-target control process is less relevant to the distance-covering phase of the movement. A larger target size is likely to require less limb-target control.

6.2.2 Dependent variables. We hypothesize that both *UStart* and *UPeak* will lead to shorter completion time and cursor total distances, which are the two dependent variables of this study, than the *PR* technique.

- Completion time: the time between selection initiation to its completion.
- Cursor total distance: the total distance traveled by the cursor from selection initiation to its completion.

6.2.3 Trial sequence. The study comprises 12 experimental conditions (= 3 techniques × 2 cluster densities × 2 task difficulties), each repeated 9 times. The cluster densities are nested within the techniques, and the task difficulties are further nested within the cluster densities. The order of all the factors is randomized. We also randomly assign the target position within a cluster after cluster expansion. This prevents participants from predicting and preplanning for the "new" target location before cluster expansion and moving directly towards there. Before starting each technique, participants are explicitly informed about how it works (i.e., the cluster expands when or before the cursor reaches it) and practiced 9 trials with $ID = 3.5$ ($D = 8$ and $W = 0.78$).

6.3 Analysis and Results

We gathered 3665 trials of data (= 34 participants × 3 techniques × 2 cluster densities × 2 task difficulties × 9 repetitions − 7 missing trials). Based on the results from the demographic questionnaire, the study sample consisted of 13 women and 21 men. The mean age of the participants was 32.1 years (std = 7.6). No trial was removed based on our data exclusion criteria. The analysis results are summarized in Table 3 and Figure 8.

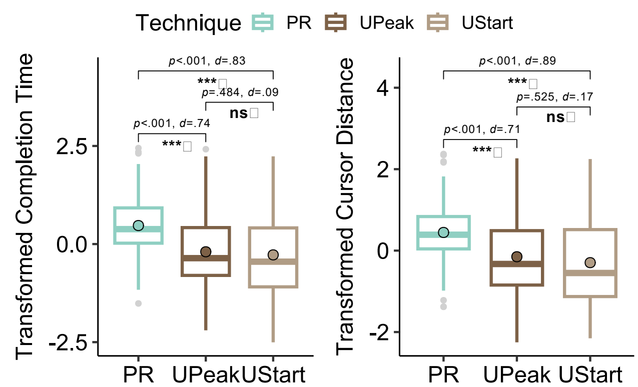

Figure 8: Boxplots display the completion time and the total cursor distance across Technique. The colored dots with a black outline indicate the mean value. Both completion time and total cursor distance were transformed with Yeo-Johnson transformations (corresponding $\lambda = -1.928$ and -4.751). Statistical significance is denoted by p-values derived from pairwise comparisons, with Cohen's d values also reported. The following symbols are used to indicate significance levels: ns: $p > 0.05$ and ***: $p \leq 0.001$.

6.3.1 Cluster expansion time. The cluster expanded an average of 0, 0.12, and 0.54 seconds (std = 0, 0.03, and 0.23 seconds) after selection initiation for *UStart*, *UPeak*, and *PR*, respectively. Note that the selection initiation timestamp was approximated based on when the cursor moved outside of the starting circle, which occurred after the actual movement initiation—this should lead to a longer actual cluster expansion time.

6.3.2 Completion time. The completion time data were transformed through a Yeo-Johnson transformation ($\lambda = -1.928$). Results from RM-ANOVA showed a main effect of Technique ($p < .001$). The post-hoc analysis suggested that *UStart* and *UPeak* led to much shorten completion time than *PR* (both $p < .001$), with no significant difference between *UStart* and *UPeak* ($p = .484$). Such a pattern was consistent across different levels of cluster densities

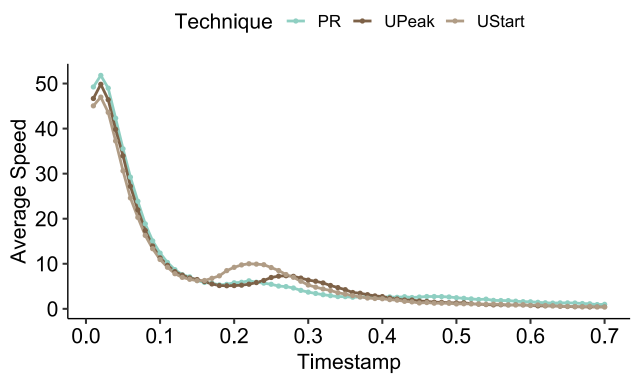

Figure 9: Lineplot of the average cursor movement speed between 0 and 0.7 seconds after exiting the starting circle.

and task difficulties, as no significant interaction effect of Technique × Density or Technique × Difficulty was identified.

6.3.3 Cursor total distance. The cursor total distance data were transformed through a Yeo-Johnson transformation ($\lambda = -4.751$). The statistical results were consistent with those of completion time.

6.3.4 Speed profile analysis. Figure 9 presents the average cursor movement speed across all trials of a technique between 0 and 0.7 seconds after exiting the starting circle. *UPeak* and *UStart* featured salient secondary peaks resulting from the limb-target control process to redirect one's movement. The redirection of *UStart* was earlier than *UPeak*. The secondary selection for *PR* was more spread out, with users tending to select expanded targets at lower speeds.

6.4 Summary

The study shows that *Unfurling*, which expands object clusters during selection movements, can reduce completion time and cursor distance significantly as compared to a progressive refinement (*PR*) technique. This finding is consistent across the tested levels of task difficulty and cluster density. The performance difference between expanding the object cluster at the start of the movement (*UStart*) versus right after reaching peak velocity (*UPeak*) is not apparent, while they differ in the resulting speed profiles.

7 Discussion

We first discuss the design implications based on the results from the three studies. We then reflect on our approach to derive techniques from theory, and introduce this method as *justified concepts*.

7.1 Selectively Hiding the Cursor

The *Cloaking* technique, which hides the cursor when selection indication feedback is enabled, can facilitate faster selection than always showing the cursor. Specifically, it works when the indication feedback is displayed before the cursor reaches the target but not when the cursor has already entered the target. This contradicts the common heuristic in HCI of always giving feedback, but aligns with our theoretical assumption that when the indication feedback is displayed beforehand, users could be lingering between the two strategies of placing the cursor within the target as pre-planned or triggering the selection without moving further. This decision prolongs their response time, and therefore the task completion time, compared to only showing the indication feedback (i.e., *Cloaking*).

However, when the cursor has entered the target, *Cloaking* does not reduce response or task completion time. This could mean that users do not pre-plan for hitting a specific location within the target but only check whether the cursor is within the target. Therefore, both the cursor-within-target and the indication feedback share the same purpose: demanding the user to trigger the selection as fast as possible. Because users can rely entirely on either of the visual feedback to complete the selection optimally, whether the cursor is hidden or not makes little difference.

The *Cloaking* technique could benefit GUIs that leverage target indication feedback to improve performance, for instance, when a predictive system can accurately provide early indication feedback of the user's intended target [41, 81, 85]. Moreover, the technique is likely to improve selection speed when the effective target boundary is larger than the visual boundary of the target [34, 76, 83]. Notably, this performance improvement compromises users' ability to accurately perceive the cursor's position when indication feedback is enabled, which may affect how users select among multiple targets, especially for relative pointing devices such as a mouse. Future research could examine how *Cloaking* works for conditions with dense targets.

7.2 Snapping the Cursor Position When Selection Begins

The *Pulsing* technique, which snaps the cursor along the movement direction at the beginning of a selection movement, can shorten the selection distance seamlessly. *Pulsing-25*, which skips about 1/4 of the original movement distance, is shown to reduce the overall movement distance without increasing the time needed for explicit corrections and does not significantly deviate from regular pointing in cursor speed profiles. It is empirically demonstrated to be a promising method to reduce movement distance with little cost from the user's side, as the impulse control process could "automatically" correct the errors induced by the snap. Yet, it contradicts the intuition that snapping further is more beneficial. Increasing the distance covered by the snap is indeed not always advantageous with this technique, as it may require additional explicit corrections. This is evident from the results of techniques that aim to reduce the initial distance by 1/2 (*Pulsing-50*) and 3/4 (*Pulsing-75*).

While *Pulsing-25* could reduce the cursor movement distance compared to regular pointing, its benefit of shortening the task completion time is not obvious (1.017 seconds *vs.* 1.024 seconds on average). It could be because the impulse control process makes subtle adjustments to the movement, resulting in a slight decrease in the overall speed of the initial impulse, as indicated by the speed profiles (Figure 6). It could also be because the mouse is a low-effort, fast input device, making the effect of *Pulsing-25* on reducing movement time less noticeable. Thus, future research could evaluate *Pulsing-25* with more cumbersome input modalities, such as bare-hand 3D interactions in mixed reality systems [2]. Additionally, it would be interesting to analyze how other transfer functions that modify speed or position of the cursor (e.g., [12, 17]) may be affected by the impulse control process.

7.3 Expanding Clustered Objects Early

The *Unfurling* technique, which expands a cluster of objects early in a selection movement, can decrease selection time and cursor movement distance compared to a typical progressive refinement technique, which requires multiple consecutive selections to narrow down the final target. This finding suggests that the limb-target control process can help reduce the discrepancy between the cursor and the target positions during a selection movement. No significant performance difference was observed between *Unfurling-Start*, which expands the cluster at the beginning of the movement, and *Unfurling-Peak*, which expands the cluster at the peak velocity of the movement impulse. Thus, a GUI can utilize a predictive system that expands the target cluster later in the movement once more information is available for prediction (e.g., after the peak velocity) to bring meaningful performance improvement [81]. This makes *Unfurling* applicable to different types of GUIs, because predicting the intended cluster for *Unfurling-Start* requires, for instance, eye-tracking, whereas the peak velocity for *Unfurling-Peak* is always available to be detected from cursor movements.

Thus, when designing a technique for selecting an object within a cluster, one might consider applying *Unfurling* to improve selection efficiency. However, there are a few practical issues that need to be accounted for. First, dynamically expanding the target could introduce more clutter in a complex interface. Therefore, it might introduce additional distractions if the intended cluster cannot be predicted accurately. Second, in a real-world application like visual menus [4], each menu item might be complex and contain text information, unlike our study, where distinct colors distinguish the target from the distractors. It is unclear whether users can effectively process and redirect their movements to items that are not easily distinguishable from the distractors. Additionally, in such applications, with sufficient practice and learning, users might memorize the target location after expansion, potentially increasing their redirection speed (i.e., their pre-planning will be influenced by using the technique). Future research could test *Unfurling* in the aforementioned scenarios, which differ from our study settings.

7.4 Deriving Techniques from Theory

We next reflect on how we derived selection techniques (i.e., design instances) from the multiple-process model (i.e., a theory). Motor control theories aim to understand and predict how selection happens, while interaction techniques in HCI aim to improve selection. Because of the distinct aims, it is non-trivial to derive techniques directly from theory.

In this research, we have managed to derive selection techniques from the multiple processes model through a set of falsifiable assumptions that are justified based on theories (we name them *justified concepts*). To be clear, the justified concepts in this work are

- Study 1: Hiding the cursor when selection indication feedback is enabled can enable faster selection than showing the cursor all the time.
- Study 2: Introducing a cursor snap at the beginning of a selection movement can shorten the cursor movement distance, therefore, the selection time seamlessly.
- Study 3: Expanding an object cluster during a selection movement allows users to redirect their ongoing movement towards the intended target, which can reduce their selection time and cursor movement distance.

These justified concepts seem to operate at a level of abstraction higher than specific design instances—one could imagine different ways to implement the concepts than *Cloaking*, *Pulsing*, and *Unfurling*, such as by using different criteria to snap the cursor or expand a target cluster. Similar to a theory, justified concepts can be falsified and iterated upon (e.g., by introducing boundary conditions [16], such as those suggested for future research above). However, they are more specific and lack the generality of theories.

Höök and Löwgren consider knowledge that resides in the middle territory between design instances and theories intermediate-level knowledge [43]. Justified concepts seem to be a type of intermediate-level knowledge but have distinct features compared to others. Justified concepts are strictly theory-grounded, unlike design heuristics that rely on one's judgment on the quality of a design [54]. We do not consider justified concepts design guidelines [48] because justified concepts are falsifiable and may therefore be wrong. Unlike strong concepts [43], justified concepts are neither design elements nor abstracted from design instances.

In addition to their uniqueness in formulation, justified concepts also seem to have distinct characteristics in their evaluation. To validate justified concepts, we first developed techniques (i.e., design instances) that carefully follow the concept. We then pre-registered the justified concepts, the technique, and the evaluation methodology before conducting the user studies. The reason was that we did not aim to generate new hypotheses as in exploratory research but to verify whether the initial logic (i.e., the justified concepts) actually applies to the technique as in confirmation research. Pre-registration required us to carefully elaborate on our arguments in advance and helped distinguish findings from post-diction and prediction [56].

Based on the study results, an empirically working technique brought us more confidence in the validity of the justified concepts. A technique that did not work as expected also offered lessons on how the theoretical derivations might have failed to work together, or the justified concepts' boundary conditions (e.g., under which scenarios the concepts do not apply) [16]. For example, we learned that *Cloaking* is effective only when the indication feedback is provided before the cursor has reached the target, which motivated us to refine the concept. Thus, we believe both expected and null results can pave the way towards more theory-grounded interaction techniques and concepts that remain interesting to iterate upon, particularly in terms of their boundary conditions. In this work, we have demonstrated three examples of how we constructed justified concepts and techniques and evaluated and discussed them, which we hope could inspire future endeavors that wish to apply a similar method.

8 Conclusion and Future Work

This paper has presented three theory-grounded concepts and selection techniques. Results from crowdsourcing-based pointing studies show that *Cloaking*, which hides the cursor when selection indication feedback is enabled, can reduce feedback response time and task completion time. *Pulsing*, which introduces a short snap at the beginning of a selection movement, can shorten the cursor movement distance seamlessly. *Unfurling*, which expands clustered objects early in the movement to allow on-the-fly redirection towards the final target, can decrease cursor movement distance and task completion time. We have discussed the design implications for GUIs based on our assumptions and empirical findings. Lastly, we reflected upon our approach to derive these techniques from theory through justified concepts, which are falsifiable assumptions justified based on existing theories. We have detailed our methodology, which involves pre-registration, to validate such concepts and techniques. Future research can evaluate and apply our concepts and techniques in various contexts, such as mixed reality systems, and in more complex task settings involving additional distractors [13, 29]. We also expect that our method of deriving interaction techniques from theory using justified concepts will inspire future endeavors.

Acknowledgments

This work was supported by the Pioneer Centre for AI, DNRF grant number P1, and Independent Research Fund Denmark (Danmarks Frie Forskningsfond) 1134-00012B.

References

[1] Motoyuki Akamatsu, I Scott MacKenzie, and Thierry Hasbroucq. 1995. A comparison of tactile, auditory, and visual feedback in a pointing task using a mouse-type device. *Ergonomics* 38, 4 (1995), 816–827. https://doi.org/10.1080/00140139508925152

[2] Ferran Argelaguet and Carlos Andujar. 2013. A survey of 3D object selection techniques for virtual environments. *Computers & Graphics* 37, 3 (2013), 121–136. https://doi.org/10.1016/j.cag.2012.12.003

[3] Felipe Bacim, Regis Kopper, and Doug A Bowman. 2013. Design and evaluation of 3D selection techniques based on progressive refinement. *International Journal of Human-Computer Studies* 71, 7-8 (2013), 785–802. https://doi.org/10.1016/j.ijhcs.2013.03.003

[4] Gilles Bailly, Eric Lecolinet, and Laurence Nigay. 2016. Visual Menu Techniques. *ACM Comput. Surv.* 49, 4, Article 60 (dec 2016), 41 pages. https://doi.org/10.1145/3002171

[5] Ravin Balakrishnan. 2004. "Beating" Fitts' law: virtual enhancements for pointing facilitation. *International Journal of Human-Computer Studies* 61, 6 (2004), 857–874. https://doi.org/10.1016/j.ijhcs.2004.09.002

[6] Marc Baloup, Thomas Pietrzak, and Géry Casiez. 2019. RayCursor: A 3D Pointing Facilitation Technique based on Raycasting. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300331

[7] Patrick Baudisch, Edward Cutrell, Mary Czerwinski, Daniel C Robbins, Peter Tandler, Benjamin B Bederson, and Alex Zierlinger. 2003. Drag-and-Pop and Drag-and-Pick: Techniques for Accessing Remote Screen Content on Touch-and Pen-Operated Systems.. In *Interact*, Vol. 3. 57–64.

[8] Patrick Baudisch, Edward Cutrell, Ken Hinckley, and Adam Eversole. 2005. Snap-and-go: helping users align objects without the modality of traditional snapping. 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, 301–310. https://doi.org/10.1145/1054972.1055014

[9] WDA Beggs and CI Howarth. 1970. Movement control in a repetitive motor task. *Nature* 225, 5234 (1970), 752–753. https://doi.org/10.1038/225752a0

[10] WDA Beggs and CI Howarth. 1972. The accuracy of aiming at a target: Some further evidence for a theory of intermittent control. *Acta psychologica* 36, 3 (1972), 171–177. https://doi.org/10.1016/0001-6918(72)90001-7

[11] Eric A. Bier. 1990. Snap-dragging in three dimensions. *SIGGRAPH Comput. Graph.* 24, 2 (feb 1990), 193–204. https://doi.org/10.1145/91394.91446

[12] Renaud Blanch, Yves Guiard, and Michel Beaudouin-Lafon. 2004. Semantic pointing: improving target acquisition with control-display ratio adaptation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vienna, Austria) *(CHI '04)*. Association for Computing Machinery, New York, NY, USA, 519–526. https://doi.org/10.1145/985692.985758

[13] 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

[14] Julien Brière and Luc Proteau. 2011. Automatic movement error detection and correction processes in reaching movements. *Experimental brain research* 208 (2011), 39–50. https://doi.org/10.1007/s00221-010-2458-1

[15] James J Burkitt, Raoul M Bongers, Digby Elliott, Steve Hansen, and James L Lyons. 2017. Extending energy optimization in goal-directed aiming from movement kinematics to joint angles. *Journal of Motor Behavior* 49, 2 (2017), 129–140. https://doi.org/10.1080/00222895.2016.1161592

[16] Christian Busse, Andrew P Kach, and Stephan M Wagner. 2017. Boundary conditions: What they are, how to explore them, why we need them, and when to consider them. *Organizational Research Methods* 20, 4 (2017), 574–609. https://doi.org/10.1177/109442811664119

[17] Géry Casiez, Daniel Vogel, Ravin Balakrishnan, and Andy Cockburn. 2008. The impact of control-display gain on user performance in pointing tasks. *Human–computer interaction* 23, 3 (2008), 215–250.

[18] Aldrich Clarence, Jarrod Knibbe, Maxime Cordeil, and Michael Wybrow. 2021. Unscripted retargeting: Reach prediction for haptic retargeting in virtual reality. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE, 150–159. https://doi.org/10.1109/VR50410.2021.00036

[19] Jacob Cohen. 2013. *Statistical power analysis for the behavioral sciences.* routledge.

[20] Carl F Craver. 2006. When mechanistic models explain. *Synthese* 153, 3 (2006), 355–376. https://doi.org/10.1007/s11229-006-9097-x

[21] Edward Robert FW Crossman and PJ Goodeve. 1963/1983. Feedback control of hand-movement and Fitts' Law. *The Quarterly Journal of Experimental Psychology* 35, 2 (1963/1983), 251–278. https://doi.org/10.1080/14640748308402133

[22] Digby Elliott and Fran Allard. 1985. The utilization of visual feedback information during rapid pointing movements. *The Quarterly Journal of Experimental Psychology Section A* 37, 3 (1985), 407–425. https://doi.org/10.1080/14640748508400942

[23] Digby Elliott, Steve Hansen, Lawrence EM Grierson, James Lyons, Simon J Bennett, and Spencer J Hayes. 2010. Goal-directed aiming: two components but multiple processes. *Psychological bulletin* 136, 6 (2010), 1023. https://doi.org/10.1037/a0020958

[24] Digby Elliott, Steven Hansen, Jocelyn Mendoza, and Luc Tremblay. 2004. Learning to optimize speed, accuracy, and energy expenditure: a framework for understanding speed-accuracy relations in goal-directed aiming. *Journal of motor behavior* 36, 3 (2004), 339–351. https://doi.org/10.3200/JMBR.36.3.339-351

[25] Digby Elliott, Werner F Helsen, and Romeo Chua. 2001. A century later: Woodworth's (1899) two-component model of goal-directed aiming. *Psychological bulletin* 127, 3 (2001), 342. https://doi.org/10.1037/0033-2909.127.3.342

[26] Digby Elliott, James Lyons, Spencer J Hayes, James J Burkitt, James W Roberts, Lawrence EM Grierson, Steve Hansen, and Simon J Bennett. 2017. The multiple process model of goal-directed reaching revisited. *Neuroscience & Biobehavioral Reviews* 72 (2017), 95–110. https://doi.org/10.1016/j.neubiorev.2016.11.016

[27] Sascha E Engelbrecht, Neil E Berthier, and Laura P O'Sullivan. 2003. The undershoot bias: learning to act optimally under uncertainty. *Psychological Science* 14, 3 (2003), 257–261. https://doi.org/10.1111/1467-9280.0343

[28] A Aldo Faisal, Luc PJ Selen, and Daniel M Wolpert. 2008. Noise in the nervous system. *Nature reviews neuroscience* 9, 4 (2008), 292–303. https://doi.org/10.1038/nrn2258

[29] Leah Findlater, Alex Jansen, Kristen Shinohara, Morgan Dixon, Peter Kamb, Joshua Rakita, and Jacob O. Wobbrock. 2010. Enhanced area cursors: reducing fine pointing demands for people with motor impairments. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology* (New York, New York, USA) *(UIST '10)*. Association for Computing Machinery, New York, NY, USA, 153–162. https://doi.org/10.1145/1866029.1866055

[30] Florian Fischer, Arthur Fleig, Markus Klar, and Jörg Müller. 2022. Optimal Feedback Control for Modeling Human–Computer Interaction. *ACM Trans. Comput.-Hum. Interact.* 29, 6, Article 51 (nov 2022), 70 pages. https://doi.org/10.1145/3524122

[31] 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. https://doi.org/10.1037/h0055392

[32] Jason P Gallivan, Craig S Chapman, Daniel M Wolpert, and J Randall Flanagan. 2018. Decision-making in sensorimotor control. *Nature Reviews Neuroscience* 19, 9 (2018), 519–534. https://doi.org/10.1038/s41583-018-0045-9

[33] Lawrence EM Grierson and Digby Elliott. 2009. Goal-directed aiming and the relative contribution of two online control processes. *The American Journal of*

Psychology 122, 3 (2009), 309–324. https://doi.org/10.2307/27784405
[34] 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
[35] Tovi Grossman and Ravin Balakrishnan. 2006. The design and evaluation of selection techniques for 3D volumetric displays. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology* (Montreux, Switzerland) *(UIST '06)*. Association for Computing Machinery, New York, NY, USA, 3–12. https://doi.org/10.1145/1166253.1166257
[36] Yves Guiard, Renaud Blanch, and Michel Beaudouin-Lafon. 2004. Object pointing: a complement to bitmap pointing in GUIs. In *Proceedings of Graphics Interface 2004* (London, Ontario, Canada) *(GI '04)*. Canadian Human-Computer Communications Society, Waterloo, CAN, 9–16.
[37] Maxime Guillon, François Leitner, and Laurence Nigay. 2015. Investigating Visual Feedforward for Target Expansion Techniques. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 2777–2786. https://doi.org/10.1145/2702123.2702375
[38] Adrian M Haith, Jina Pakpoor, and John W Krakauer. 2016. Independence of movement preparation and movement initiation. *Journal of Neuroscience* 36, 10 (2016), 3007–3015. https://doi.org/10.1523/JNEUROSCI.3245-15.2016
[39] Steve Hansen, Cheryl M Glazebrook, J Greg Anson, Daniel J Weeks, and Digby Elliott. 2006. The influence of advance information about target location and visual feedback on movement planning and execution. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale* 60, 3 (2006), 200. https://doi.org/10.1037/cjep2006019
[40] Matthew Heath, Nicola J Hodges, Romeo Chua, and Digby Elliott. 1998. On-line control of rapid aiming movements: Unexpected target perturbations and movement kinematics. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale* 52, 4 (1998), 163. https://doi.org/10.1037/h0087289
[41] Rorik Henrikson, Tovi Grossman, Sean Trowbridge, Daniel Wigdor, and Hrvoje Benko. 2020. Head-Coupled Kinematic Template Matching: A Prediction Model for Ray Pointing in VR. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376489
[42] Ken Hinckley and Daniel Wigdor. 2013. Input Technologies and Techniques. *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications, Third Edition* (2013), 95–132.
[43] Kristina Höök and Jonas Löwgren. 2012. Strong concepts: Intermediate-level knowledge in interaction design research. *ACM Trans. Comput.-Hum. Interact.* 19, 3, Article 23 (oct 2012), 18 pages. https://doi.org/10.1145/2362364.2362371
[44] Kasper Hornbæk. 2006. Current practice in measuring usability: Challenges to usability studies and research. *International journal of human-computer studies* 64, 2 (2006), 79–102. https://doi.org/10.1016/j.ijhcs.2005.06.002
[45] Steven W Keele. 1968. Movement control in skilled motor performance. *Psychological bulletin* 70, 6p1 (1968), 387. https://doi.org/10.1037/h0026739
[46] Mikko Kytö, Barrett Ens, Thammathip Piumsomboon, Gun A. Lee, and Mark Billinghurst. 2018. Pinpointing: Precise Head- and Eye-Based Target Selection for Augmented Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3173574.3173655
[47] Byungjoo Lee, Mathieu Nancel, Sunjun Kim, and Antti Oulasvirta. 2020. AutoGain: Gain Function Adaptation with Submovement Efficiency Optimization. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376244
[48] William Lidwell, Kritina Holden, and Jill Butler. 2010. *Universal principles of design, revised and updated: 125 ways to enhance usability, influence perception, increase appeal, make better design decisions, and teach through design.* Rockport Pub.
[49] Christine L MacKenzie, RG Marteniuk, C Dugas, D Liske, and B Eickmeier. 1987. Three-dimensional movement trajectories in Fitts' task: Implications for control. *The Quarterly Journal of Experimental Psychology* 39, 4 (1987), 629–647. https://doi.org/10.1080/14640748708401806
[50] Michael McGuffin and Ravin Balakrishnan. 2002. Acquisition of expanding targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Minneapolis, Minnesota, USA) *(CHI '02)*. Association for Computing Machinery, New York, NY, USA, 57–64. https://doi.org/10.1145/503376.503388
[51] 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
[52] David E Meyer, JE Keith-Smith, Sylvan Kornblum, Richard A Abrams, and Charles E Wright. 1990. Speed-accuracy tradeoffs in aimed movements: Toward a theory of rapid voluntary action. (1990).
[53] Mark R Mine. 1995. Virtual environment interaction techniques. *UNC Chapel Hill CS Dept* (1995).
[54] Jakob Nielsen and Rolf Molich. 1990. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Seattle, Washington, USA) *(CHI '90)*. Association for Computing Machinery, New York, NY, USA, 249–256. https://doi.org/10.1145/97243.97281
[55] Karin Nieuwenhuizen, Dzmitry Aliakseyeu, and Jean-Bernard Martens. 2010. Insight into goal-directed movement strategies. 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, 883–886. https://doi.org/10.1145/1753326.1753457
[56] Brian A Nosek, Charles R Ebersole, Alexander C DeHaven, and David T Mellor. 2018. The preregistration revolution. *Proceedings of the National Academy of Sciences* 115, 11 (2018), 2600–2606. https://doi.org/10.1073/pnas.1708274114
[57] Phillip T. Pasqual and Jacob O. Wobbrock. 2014. Mouse pointing endpoint prediction using kinematic template matching. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 743–752. https://doi.org/10.1145/2556288.2557406
[58] Y Paulignan, Marc Jeannerod, C MacKenzie, and R Marteniuk. 1991. Selective perturbation of visual input during prehension movements: 2. The effects of changing object size. *Experimental brain research* 87 (1991), 407–420. https://doi.org/10.1007/BF00231858
[59] Y Paulignan, C MacKenzie, R Marteniuk, and Marc Jeannerod. 1991. Selective perturbation of visual input during prehension movements: 1. The effects of changing object position. *Experimental brain research* 83 (1991), 502–512. https://doi.org/10.1007/BF00229827
[60] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. 1996. The go-go interaction technique: non-linear mapping for direct manipulation in VR. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology* (Seattle, Washington, USA) *(UIST '96)*. Association for Computing Machinery, New York, NY, USA, 79–80. https://doi.org/10.1145/237091.237102
[61] Claude Prablanc and Olivier Martin. 1992. Automatic control during hand reaching at undetected two-dimensional target displacements. *Journal of neurophysiology* 67, 2 (1992), 455–469. https://doi.org/10.1152/jn.1992.67.2.455
[62] Luc Proteau and Guillaume Masson. 1997. Visual perception modifies goal-directed movement control: Supporting evidence from a visual perturbation paradigm. *The Quarterly Journal of Experimental Psychology Section A* 50, 4 (1997), 726–741. https://doi.org/10.1080/713755729
[63] Roger Ratcliff and Gail McKoon. 2008. The diffusion decision model: theory and data for two-choice decision tasks. *Neural computation* 20, 4 (2008), 873–922. https://doi.org/10.1162/neco.2008.12-06-420
[64] David A Rosenbaum. 2009. *Human motor control.* Academic press.
[65] Rufat Rzayev, Sven Mayer, Christian Krauter, and Niels Henze. 2019. Notification in VR: The Effect of Notification Placement, Task and Environment. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play* (Barcelona, Spain) *(CHI PLAY '19)*. Association for Computing Machinery, New York, NY, USA, 199–211. https://doi.org/10.1145/3311350.3347190
[66] Fabrice Sarlegna, Jean Blouin, Jean-Louis Vercher, Jean-Pierre Bresciani, Christophe Bourdin, and Gabriel M Gauthier. 2004. Online control of the direction of rapid reaching movements. *Experimental Brain Research* 157 (2004), 468–471. https://doi.org/10.1007/s00221-004-1860-y
[67] Fabrice R Sarlegna and Pratik K Mutha. 2015. The influence of visual target information on the online control of movements. *Vision research* 110 (2015), 144–154. https://doi.org/10.1016/j.visres.2014.07.001
[68] Jeffrey A Saunders and David C Knill. 2003. Humans use continuous visual feedback from the hand to control fast reaching movements. *Experimental brain research* 152 (2003), 341–352. https://doi.org/10.1007/s00221-003-1525-2
[69] Abraham Savitzky and Marcel JE Golay. 1964. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry* 36, 8 (1964), 1627–1639. https://doi.org/10.1021/ac60214a047
[70] Richard A Schmidt, Howard Zelaznik, Brian Hawkins, James S Frank, and John T Quinn Jr. 1979. Motor-output variability: a theory for the accuracy of rapid motor acts. *Psychological review* 86, 5 (1979), 415. https://doi.org/10.1037/0033-295X.86.5.415
[71] Stephen H Scott. 2016. A functional taxonomy of bottom-up sensory feedback processing for motor actions. *Trends in neurosciences* 39, 8 (2016), 512–526. https://doi.org/10.1016/j.tins.2016.06.001
[72] Jeroen BJ Smeets, Leonie Oostwoud Wijdenes, and Eli Brenner. 2016. Movement adjustments have short latencies because there is no need to detect anything. *Motor Control* 20, 2 (2016), 137–148. https://doi.org/10.1123/mc.2014-0064
[73] 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
[74] Ivan E. Sutherland. 1964. Sketch pad a man-machine graphical communication system. In *Proceedings of the SHARE Design Automation Workshop (DAC '64)*. Association for Computing Machinery, New York, NY, USA, 6.329–6.346. https:

//doi.org/10.1145/800265.810742

[75] Wen-Jie Tseng, Petros Dimitrios Kontrazis, Eric Lecolinet, Samuel Huron, and Jan Gugenheimer. 2024. Understanding Interaction and Breakouts of Safety Boundaries in Virtual Reality Through Mixed-Method Studies. In *2024 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*. IEEE, 482–492. https://doi.org/10.1109/VR58804.2024.00069

[76] Hiroki Usuba, Shota Yamanaka, and Homei Miyashita. 2021. Modeling Movement Times and Success Rates for Acquisition of One-dimensional Targets with Uncertain Touchable Sizes. *Proc. ACM Hum.-Comput. Interact.* 5, ISS, Article 487 (nov 2021), 15 pages. https://doi.org/10.1145/3486953

[77] Mélodie Vidal, Andreas Bulling, and Hans Gellersen. 2013. Pursuits: spontaneous interaction with displays based on smooth pursuit eye movement and moving targets. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Zurich, Switzerland) *(UbiComp '13)*. Association for Computing Machinery, New York, NY, USA, 439–448. https://doi.org/10.1145/2493432.2493477

[78] Timothy N Welsh, Laura Higgins, and Digby Elliott. 2007. Are there age-related differences in learning to optimize speed, accuracy, and energy expenditure? *Human Movement Science* 26, 6 (2007), 892–912. https://doi.org/10.1016/j.humov.2007.04.004

[79] Robert Sessions Woodworth. 1899. Accuracy of voluntary movement. *The Psychological Review: Monograph Supplements* 3, 3 (1899), i.

[80] Difeng Yu, Mantas Cibulskis, Erik Skjoldan Mortensen, Mark Schram Christensen, and Joanna Bergström. 2024. Metrics of Motor Learning for Analyzing Movement Mapping in Virtual Reality. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 724, 18 pages. https://doi.org/10.1145/3613904.3642354

[81] Difeng Yu, Ruta Desai, Ting Zhang, Hrvoje Benko, Tanya R. Jonker, and Aakar Gupta. 2022. Optimizing the Timing of Intelligent Suggestion in Virtual Reality. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) *(UIST '22)*. Association for Computing Machinery, New York, NY, USA, Article 6, 20 pages. https://doi.org/10.1145/3526113.3545632

[82] Difeng Yu, Tilman Dingler, Eduardo Velloso, and Jorge Goncalves. 2024. Object Selection and Manipulation in VR Headsets: Research Challenges, Solutions, and Success Measurements. *ACM Comput. Surv.* 57, 4, Article 98 (Dec. 2024), 34 pages. https://doi.org/10.1145/3706417

[83] Difeng Yu, Hai-Ning Liang, Xueshi Lu, Kaixuan Fan, and Barrett Ens. 2019. Modeling endpoint distribution of pointing selection tasks in virtual reality environments. *ACM Trans. Graph.* 38, 6, Article 218 (nov 2019), 13 pages. https://doi.org/10.1145/3355089.3356544

[84] Difeng Yu, Qiushi Zhou, Joshua Newn, Tilman Dingler, Eduardo Velloso, and Jorge Goncalves. 2020. Fully-occluded target selection in virtual reality. *IEEE transactions on visualization and computer graphics* 26, 12 (2020), 3402–3413. https://doi.org/10.1109/TVCG.2020.3023606

[85] Shumin Zhai, Carlos Morimoto, and Steven Ihde. 1999. Manual and gaze input cascaded (MAGIC) pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Pittsburgh, Pennsylvania, USA) *(CHI '99)*. Association for Computing Machinery, New York, NY, USA, 246–253. https://doi.org/10.1145/302979.303053