Empirical Evaluation of Moving Target Selection in Virtual Reality Using Egocentric Metaphors

Yuan Chen[1,2](✉), Junwei Sun[2], Qiang Xu[2], Edward Lank[1], Pourang Irani[3], and Wei Li[2]

[1] University of Waterloo, Waterloo, ON, Canada
{y2238che,lank}@uwaterloo.ca
[2] Huawei Technologies, Markham, ON, Canada
{junwei.sun,qiang.xu1,wei.li.crc}@huawei.com
[3] University of Manitoba, Winnipeg, MB, Canada
irani@cs.umanitoba.ca

Abstract. Virtual hand or pointer metaphors are among the key approaches for target selection in immersive environments. However, targeting moving objects is complicated by factors including target speed, direction, and depth, such that a basic implementation of these techniques might fail to optimize user performance. We present results of two empirical studies comparing characteristics of virtual hand and pointer metaphors for moving target acquisition. Through a first study, we examine the impact of depth on users' performance when targets move beyond and within arms' reach. We find that movement in depth has a great impact on both metaphors. In a follow-up study, we design a reach-bounded Go-Go (rbGo-Go) technique to address challenges of virtual hand and compare it to Ray-Casting. We find that target width and speed are significant determinants of user performance and we highlight the pros and cons for each of the techniques in the given context. Our results inform the UI design for immersive selection of moving targets.

1 Introduction

Selection of moving targets is a common interaction in video games and when inspecting surveillance videos [21]. Moving target acquisition is challenging as it requires users to predict the speed and trajectory of objects. Furthermore, if the object is moving in depth, i.e. towards or away from users, users may also need to accurately predict its depth. In addition, acquiring moving targets in virtual

Y. Chen—The work was done when the author was an intern at Huawei Canada.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-85610-6_3) contains supplementary material, which is available to authorized users.

C. Ardito et al. (Eds.): INTERACT 2021, LNCS 12935, pp. 29–50, 2021.
https://doi.org/10.1007/978-3-030-85610-6_3

environments lacks the feedback when commonly interacting with the movement of physical objects, such as catching a baseball.

With the introduction of high-performance headsets and controllers, immersive experiences in augmented reality (AR) and virtual reality (VR) are increasingly available to consumers. Pursuing immersion has always been an end-goal in AR/VR research, and interacting with moving targets – in games [53], training environments [10], and virtual worlds (e.g. Second Life) – is an important component of this goal [10,53].

Despite extensive research in target selection techniques [7,31,39,43] and the common occurrence of moving target interaction, we identify two issues with existing work. First, the majority of target selection work in 3D environments focuses on static objects, which does not directly translate to an understanding of moving target acquisition. Second, when looking specifically at AR/VR environments, techniques for pointing and selection remain clustered across two primary metaphors—the virtual hand and the virtual pointer – which have different characteristics. Poupyrev et al. [42] identified characteristics of these two metaphors for static object selection and re-positioning tasks, however we know very little about their performance in a moving target selection task.

In this paper, we empirically explore the performance of these two common selection metaphors (virtual hand and virtual pointer) when acquiring moving targets in VR. Our work consists of two studies to evaluate these selection metaphors. In the first study, participants were asked to use a 1-to-1 mapped controller with a virtual hand and with Ray-Casting to select a moving target which appeared in a limited volume within arms' reach and at a distance. We found that depth had a significant impact on both metaphors. Given that a 1-to-1 mapping limits the reachable space when using the virtual hand, in the second study we designed a technique, called reach-bounded Go-Go (rbGo-Go), an extension of the Go-Go [43] technique, to capture distant targets. We compared rbGo-Go with Ray-Casting and found that Ray-Casting was more efficient than rbGo-Go, but had similar accuracy. We found that target speed and width were dominant factors in users' performance. Our results lead us to believe that both Ray-Casting and rbGo-Go are viable alternatives with advantages and disadvantages for distant moving target selection in VR/AR contexts. We discuss these complementary advantages and disadvantages to provide guidance for designers of experiences that incorporate moving target selection in virtual environments.

In summary, the contributions in this paper include:

- Two empirical studies to evaluate the performance of virtual hand and pointer metaphors and establish baselines to capture moving targets in VR.
- A summary of technique characteristics and design considerations for both virtual hand and pointer metaphors to acquire moving targets in VR.

2 Related Work

2.1 Virtual Hand and Virtual Pointer

Poupyrev et al. [42] identified a class of manipulations labeled as egocentric manipulations of objects in virtual environments. They further divided this class of egocentric manipulations into two categories: virtual hand and virtual pointer. The virtual hand metaphor [38] and Ray-Casting [7] (a virtual pointer technique) are the most popular 3D pointing techniques within virtual environments.

Many variants of ray-casting and virtual hand techniques have been proposed for interacting with 3D objects in virtual environments (VE) [5]. Considering first, Ray-Casting, many techniques explore variants for depth-aware pointing, such as manually adjusting the length or depth of a ray [11,19], creating a curved ray [39,44] and allowing bimanual interaction between rays [52] to increase targeting precision. In dense environments, techniques such as SQUAD [31], Expand [13] and Disambiguation Canvas [16] use iterative refinement by rearranging or filtering content to support Ray-Casting selection. More recently, the RayCursor technique [7] incorporates a set of pointing facilitation augmentations including the 1€ filter [14], a bubble cursor mechanism [18] and visual feedback to achieve better performance on small and distant targets. Finally, Ray-Casting from a user's eyes [4] has been shown to have some performance advantages in both sparse and cluttered environments, and a similar technique – cursor control using head – has also been adopted in existing AR headsets, for example the Microsoft HoloLens.

In contrast to Ray-Casting, where a ray points to a target, the virtual hand metaphor allows users to directly interact with objects using their bare hands or controllers [6]. One challenge with virtual hand techniques is the acquisition of distant targets; to address this, the Go-Go technique [43] enables users to grab targets beyond arms' reach with a non-linear mapping function; close interactions leverage basic virtual hand input mapped to a controller, but, as the user reaches further than a threshold from their body, a control-display (CD) gain function magnifies the movement of the hand beyond the threshold. Although Go-Go is functional for distant targeting, it does not perform as well as Ray-Casting based techniques for object positioning [42,46]. Another body of work within the category of virtual hand metaphors for distant reaching focuses on maintaining body ownership while enabling users to reach distant targets [48].

2.2 Moving Targets Acquisition Models

Acquiring moving targets is challenging to model. When acquiring a moving target, users may use different strategies (pursuit, head-on, receding, and perpendicular) to intercept a moving object [47]. Li et al. [34] argued that different model clues (vision, haptic and audio and their combinations), motion direction, and speed could affect users' performance in VR. Fitts's Law [17] is the most commonly used approach to study new acquisition techniques in a spatial acquisition task. It links movement time (MT) to the concept of the index of difficulty (ID): MT = a + bID where $ID = log_2(\frac{2D}{W})$. However, Fitts' Law was initially proposed to model stationary target selection in 1D. While it has been extended to 2D pointing tasks [2], it is clear that extensions to Fitts' Law are needed to account for, at minimum, target speed (i.e. faster targets should be harder to acquire). With this goal in mind, Jagacinski et al. [26] extended Fitts' Law and proposed an empirical model to predict movement time as a function of initial amplitude

(A), target speed (V) and target width (W): $MT = c + dA + e(V + l)(\frac{1}{W} - 1)$. Hoffmann [22] later examined three extensions to Fitts' Law which all indicated that the difficulty to select a moving target is correlated with the target speed.

In addition to modeling the movement time, previous work also focuses on predicting the endpoint distribution and the error rate of target selection. One of the benefits to studying endpoint distribution is that it allows replacement of nominal target width with the target's effective width [55], where ideally 96% of the endpoints land within a target following a Gaussian distribution that corresponds to the target width (W) perceived by the user, i.e., $W = \sqrt{2\pi e}\sigma$ [36]. This 4% error rate assumption by Fitts' Law provides insights into adjusting W and W_e, and, based on this, Wobbrock et al. [50] derive an error model implicitly implied by Fitts' Law with three task parameters: target width, distance and movement time. They extend the error model to 2D [51] and Bi et al. also propose the FFitts' Law [9] to model the endpoint distribution of finger input on a touchscreen given the challenges of *fat finger* input. For moving targets, Lee and Oulasvirta [32] propose a statistical model to predict the error rate by considering moving target selection as a temporal pointing task where users intend to select a target within a limited time window. Several other researchers propose Ternary-Gaussian models to describe the endpoint distribution of moving targets in 1D [24], 2D [25], and with crossing-based movement [23]. Focusing specifically on VR and AR environments, Yu et al. [54] recently designed an empirical model *EDModel* to describe the distribution of pointing selection tasks in VR environments. These endpoint distribution and error rate models imply that the target width and effective target width of the selection task have more impact on the error rate of selecting moving targets than the distance.

2.3 Moving Targets Acquisition Techniques

Based on the above mentioned models, many researchers have investigated approaches to efficiently and accurately completing a stationary selection task; these facilitation techniques can be characterized into two broad categories: (1) decreasing movement distance from the cursor to the target [8], (2) increasing the effective width with larger cursor width [28,45], target width [37] or activation area [18,45]. Adopting these techniques, Hassan et al. [21] propose Comet and Target Ghost to capture moving targets in 1D and 2D space. With Comet, targets are expanded with a long tail such that the activation area is increased while Target Ghost ignores targets' speed and creates static proxies to the targets. Similarly, the Hold [3] technique temporarily pauses the content of the display to enable a static target selection. Aside from these approaches, Hook [40] adopts a vote-score heuristics designed to select a moving target in a dense and occluded environment while Khamis et al. [29] propose an eye-pursuit technique to select moving targets in a VR environment. These techniques are designed on top of basic interaction techniques, e.g. mouse cursor, Ray-Casting and eye-tracking.

Interestingly, although myriad techniques based on virtual hand and virtual pointer have been proposed for selection in virtual environments, evaluation of the performance of these two metaphors [42] has typically focused on static selection tasks. The lack of empirical understanding of the basic performance of these established techniques to acquire moving targets in virtual environments still exists and we are aware of no prior work that has explored these two metaphors in the context of moving targets in AR/VR.

3 Study Set-Up

In this section, we describe our virtual pointer, virtual hand, and configuration parameters for our moving target selection task.

3.1 Virtual Pointer

Ray-Casting is the most common technique built on top of the virtual pointer metaphor. For our virtual pointer selection technique, Ray-Casting is implemented using a constant control display (C/D) gain to map a controller orientation onto a ray direction. A maximum 10-m white line shooting from a virtual controller indicates the ray direction. When a ray intersects with an object, the white line's length is then the distance from the controller to the object and a cursor appears to indicate the hit position. We do not enable cursor manipulation and acceleration along the ray because this would shift users' attention when acquiring a moving target, possibly increasing the selection time. Approaches which can improve Ray-Casting's efficiency and stability, such as increasing the effective width [18] of a target or smoothing of the ray [14] are not implemented because these approaches will benefit Ray-Casting over the virtual hand, which violates our initial motivation towards the understanding of virtual hand in comparison to the pointer in moving target selection. A press down&up gesture on the corresponding trigger of the VR controller selects a target intersecting the ray. We call this technique **Ray-Casting** in our experimental condition.

3.2 Virtual Hand

Using bare hands or a controller to hit/select a target is the most basic form of virtual hand. The basic virtual hand metaphor allows users to acquire objects within arms' reach. While this constraint mimics the real world, researchers have also recognized that, at times, users may wish to acquire and interact with targets beyond arms' reach. As a result, modifications to the virtual hand metaphor to support more distant target acquisition exist [33,43,48].

The variant we initially explore, is a teleporting technique where the virtual hand is located at a distance from the user. The virtual hand remains directly mapped via a constant CD gain onto the controller movement. This teleporting technique permits distant targeting while preserving the virtual hand metaphor. We call this technique **Controller-mapped virtual hand** or **Controller** for our experimental condition.

3.3 Randomize Movement Direction with Bounded Space

As targets are moving, and, because users find it difficult to select targets outside the field-of-view in the virtual environment [15], we designed a bounded space which is anchored to the virtual environment and contains all moving targets in front of the user. This allows us to analyse the effects from target speed, width and technique without visual search and out-of-field-of-view confounds. The bounded space is designed based on empirical analysis of the reachable workspace within arm's length [30,41] so we can use a linear 1-to-1 mapping between the physical and virtual hand to interact with targets within the bounded space. The bounded space is 60 cm (depth) × 120 cm (height) × 120 cm (width) in size; thus enabling users to reach each side of the bounded space. To aid depth estimation and avoid visual distraction, the wall nearest to a user's chest is invisible while other walls are semi-transparent.

Alongside objects' movement range, another important factor that affects users' performance is movement direction. Prior work [47] has suggested that the direction of motion relative to the cursor can result in different targeting strategies. Instead of controlling the target motion directly, we added a bounce feature on each wall of the bounded space so that whenever a target hits a wall, it will reflect and bounce away. Therefore, target motion direction is randomized and users can leverage different strategies to capture a moving target.

3.4 Pilot Study to Guide Moving Target Speed

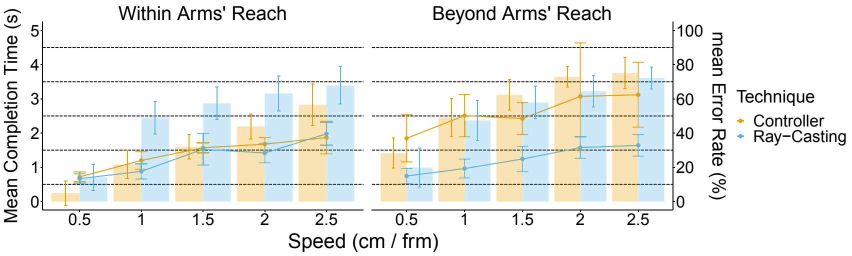


Fig. 1. Mean selection time (plotted in line) and accuracy (in bar), with 95% confidence intervals, of Controller and Ray-Casting to acquire moving targets within/beyond arms' reach at different speeds. The frame rate is 72 fps.

Higher motion speed implies longer selection time and higher error rate [22, 26,32]. Without pre-assuming the acceptable minimum and maximum speed values in the experiments, we conducted informal pilot test with 4 adults for feedback on speed with regard to selection accuracy and completion time. As the goal of our pilot study was to guide speed values, we did not vary target width. The task in the pilot study is a simple selection task where participants were expected to select the only sphere moving within and beyond arms' reach across different speeds: min = 0.5 cm/frm, max = 2.5 cm/frm, step = 0.5 cm/frm.

We used 3 Block × 2 Position × 2 Technique × 5 Speed × 10 trials in this study. Examining Fig. 1, the error rate reached 50% at 1.0 cm/frm for distant moving target selection using the virtual hand and participants reported it was too easy to select targets nearby at 0.5 cm/frm, so we selected speed values of (0.75 cm/frm, 1.00 cm/frm, 1.25 cm/frm) for target movement.

4 Initial Study

Given speed values from our pilot study, our initial experiment's goal was to explore if virtual hand (**Controller**) and virtual pointer (**Ray-Casting**) have different performance when selecting moving targets with different speeds and at different positions (Fig. 2).

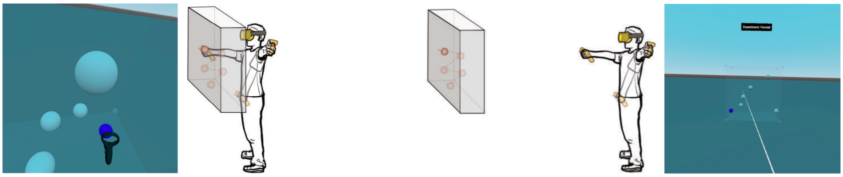


Fig. 2. Within (left) and beyond (right) arms' design: Given a user's head position, near space's center is generated at 0.3 m in front of and 0.4 m below the head position. Far space's center is 2 m beyond that of near space. These configuration values were consistent across participants.

4.1 Apparatus and Implementation

The system was implemented in Godot v3.2.2 stable and deployed on an standalone Oculus Quest at 72 fps. No other hardware resources were required.

4.2 Experimental Design

A repeated-measure within-subject design was used. The independent variables (IVs) were *Technique* (Controller, Ray-Casting),*Position* (Within, Beyond), *Speed* (0.75 cm/frm, 1.00 cm/frm, 1.25 cm/frm), and *Block* (1-4). As *Position* might affect perceived width for moving targets and to handle the confounding problem, target width was fixed to 10 cm.

Participants were instructed to capture a set of moving targets during the experiment. Spheres (targets) were generated, moving and bouncing back and forth in the bounded space. *Position* was either within arms reach or out of reach. *Technique* by *Position* generates four combinations and the order of the combination was counterbalanced across participants using a Latin square [49].

Each participant performed 4 blocks of trials. To start each block and control the initial position, each participant was instructed to select a static white sphere ("dummy" target) positioned in front of the participant in the virtual space.

Within each block, 15 selections were made for each of the 3 target motion speeds, presented in random order in each block, given the *Technique* and *Position* condition being analyzed. For each trial, six spheres, including 5 white spheres (distractors) and 1 blue sphere (goal target) were generated within the bounded space with the same *Speed* but moving in random direction. When Controller overlapped with a sphere or Ray-Casting intersected a sphere, the sphere was highlighted orange and the participant pressed the index-finger button of the corresponding controller to capture it. The experimental system moved to the next trial when the goal target was correctly selected. When a target was missed or a distractor was selected, the goal target did not disappear. Once correctly selected, the current goal target vanished and the next goal target would generate at a distance (20 cm, 30 cm, 40 cm in random order) from the selection position of the *Technique*, i.e. the Controller's position or the hit position of Ray-Casting. When each participant finished a block, the static "dummy" target was displayed and the participant could take a break. In summary, each participant performed 2 *Technique* × 2 *Position* × 3 *Speed* × 4 *Block* × 15 trials = 720 trials.

4.3 Participants

We recruited 8 participants (ages 21 to 30 ($\mu = 25.3, \sigma = 2.7$), 5 male, 3 female, 2 left-handed), of which 2 were experienced VR users. Participants were recruited by word-of-mouth in our organization. The experiment lasted for 40 min.

4.4 Procedure

Participants were welcomed to the study and were instructed to stand in a open area. They first read the study instructions and verbal consent was obtained. Before the study, they were asked to answer a questionnaire about demographic information (gender, age), handedness, and daily and weekly usage of VR devices to characterize the demographics of our participant sample. Participants were warned about potential motion sickness induced from VR, and were allowed to have a 30 s break between each block. If they felt uncomfortable at any time during the study, the study immediately stopped. Before participants wore the Oculus Quest and controllers, both the headset and controllers were sanitized with alcoholic wipes. Prior to the study, a training session was provided so participants practiced and became familiar with the provided techniques and environment. Participants were instructed to avoid continually hitting the select button. When participants finished a block, they could take a break; when they finished a *Technique* and *Position* condition, they were allowed to take off the headset and have a 3 min break and, at the same time, they were instructed to complete a raw NASA TLX [20] questionnaire grading their experience.

4.5 Results

The goal of our experiment is the contrast of virtual hand and virtual pointer metaphors for moving target selection. We acknowledge that various extensions based on Fitts's Law and endpoint prediction models have been proposed to analyze moving target selection experiments. However, in our study, the complexity of 3D targeting in the immersive environment, unpredictable movement direction caused by reflection in the bounded space, and distracting moving objects introduce confounds in analysis. Therefore, in the following sections, we analyze our results in two dimensions: (1) objective measures: selection time and error rate, and (2) subjective measures: TLX loads and user feedback. The selection time refers to the time elapsed between selections. Selection failure is counted as an error and the error rate refers to the percentage of erroneous trials among 15 trials.

We removed outliers by eliminating any non-erroneous trials whose selection time was more than three standard deviations from the mean, yielding 5684 trials (98.68%) in total for analysis. We conducted a multi-way repeated-measure ANOVA ($\alpha = 0.05$) for selection time and error rate respectively on three IVs: *Technique*, *Position*, and *Speed*. When sphericity was violated using Mauchly's test, we applied Greenhouse-Geisser corrections to the DoFs. The post-hoc tests were conducted using pairwise t-tests with Bonferroni corrections when significant effects were found. Effect sizes are reported as partial eta squared (η_p^2) values.

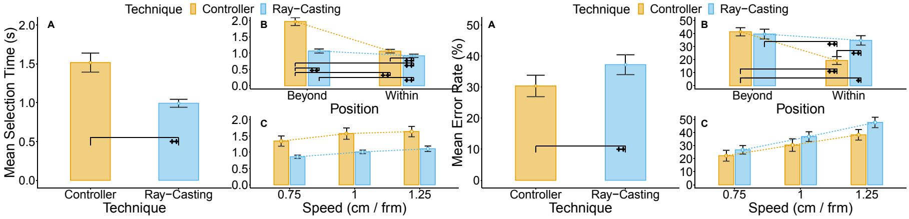


Fig. 3. Mean selection time (left) and mean error rate (right) for (A) *Technique*; (B) *Technique* by *Position*; (C) *Technique* by *Speed*. Error bars are shown with 95% confidence intervals. The statistic significance evaluated by pairwise t-test are marked with + (++ = p < 0.01 and + = p < 0.05).

Selection Time. A Box-Cox transformation ($\lambda = -0.5$) was applied to non-normal residuals of the selection time. Although we found a significant effect of *Block* ($F_{3,21} = 3.61$, $p < 0.05$, $\eta_p^2 = 0.03$), the pairwise t-test did not report any significance between each pair of block. Therefore, all 4 blocks were kept for the analysis.

The subsequent analysis revealed a significant effect of *Technique* ($F_{1,7} = 58.02$, $p < 0.001$, $\eta_p^2 = 0.45$) on selection time. The pairwise t-test showed that Controller (mean = 1.52 s) was significantly slower than Ray-Casting (0.99 s, $p < 0.001$). We found a significant effect of *Position* ($F_{1,7} = 51.40$, $p < 0.001$, $\eta_p^2 = 0.44$) on the selection time and a significant interaction effect between *Technique* and *Position* ($F_{1,7} = 44.85$, $p < 0.001$, $\eta_p^2 = 0.20$). Participants spent significantly ($p < 0.001$) more time selecting moving targets at a distance (1.52 s) than within arms' reach (0.99 s). The selection time increased significantly

($p < 0.001$) with Controller when targets appeared beyond arms' reach (1.97 s) than within arms' reach (1.06 s). Similarly, when using Ray-Casting, the selection time increased significantly ($p < 0.001$) when targets appeared far away from (1.07 s) than near participants (0.92 s). We also found a significant effect of *Speed* ($F_{2,14} = 44.74$, $p < 0.001$, $\eta_p^2 = 0.21$). Selecting targets moving at 0.75 cm/frm (1.10 s) was significantly ($p < 0.005$) faster than those moving at 1.00 cm/frm (1.29 s) and 1.25 cm/frm (1.37 s). We only found a significant interaction effect between *Speed* and *Position* ($F_{2,14} = 3.90$, $p < 0.05$, η_p^2 <0.01) but not with *Technique*. To potentially avoid a type-I error, we consider the interaction effect trivial given the effect size.

Accuracy. We did not find any significant effect of *Block* on error rate. Analysis reveals a significant effect of *Technique* ($F_{1,7} = 7.56$, $p < 0.05$, $\eta_p^2 = 0.10$) on the error rate. Controller (mean = 30.40%) caused significantly less erroneous selections than Ray-Casting (37.25%, $p < 0.001$). We also found a significant effect of *Position* ($F_{1,7} = 205.61$, $p < 0.001$, $\eta_p^2 = 0.29$) on error rate; selecting moving targets that appeared at a distance (40.57%) caused significantly more errors than within arms' reach (27.09%, $p < 0.001$). We found a significant interaction effect between *Technique* and *Position* ($F_{1,7} = 17.28$, $p < 0.005$, $\eta_p^2 = 0.15$). With Controller, participants made significantly more erroneous selection when selecting targets at a distance (41.45%) than within arms' reach (19.35%, $p < 0.001$). Ray-Casting was less impacted by distance; i.e., errors at a distance (39.68%) and within arms' reach (34.83%) were similar. We found a significant effect of *Speed* ($F_{2,14} = 52.99$, $p < 0.001$, $\eta_p^2 = 0.35$) on the error rate but we did not found any significant interaction effect between *Speed* and other IVs on the error rate. Unsurprisingly, selecting targets moving at 0.75 cm/frm (24.57%) caused significantly ($p < 0.001$) lower erroneous selection than those moving at 1.00 cm/frm (33.77%) and 1.25 cm/frm (43.14%).

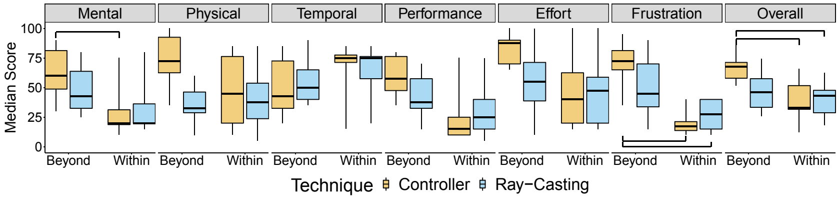


Fig. 4. Box plots for perceived task loads of the TLX questionnaire. The statistical significant differences are marked as connecting lines.

Task Loads Analysis and User Feedback. Results from Fig. 4 showed differences for perceived task loads between techniques beyond and within arms' reach. A Friedman test showed significant effect of *Technique* × *Position* on *Mental*, *Effort*, *Frustration* and *Overall*: $\chi^2_{Mental}(3) = 21.83$, $p < 0.001$, $\chi^2_{Effort}(3) = 11.95$,

$p < 0.01$, $\chi^2_{Frustration}(3) = 20.01$, $p < 0.001$, $\chi^2_{Overall}(3) = 15.83$, $p < 0.005$. The pairwise Wilcoxon test reported that using Controller to acquire moving targets beyond arms' reach caused significantly higher mental demand ($p = 0.05$), higher frustration ($p < 0.05$), and required significantly ($p < 0.05$) higher overall loads than using Ray-Casting or Controller to select moving targets within arms' reach. To explore more fully *Position*, we found a significant effect of *Position* on all attributes except *Physical*, *Temporal*, and *Performance*: $\chi^2_{Mental}(1) = 8, p < 0.005$, $\chi^2_{Effort}(1) = 6, p < 0.05$, $\chi^2_{Frustration}(1) = 8, p < 0.005$, $\chi^2_{Overall}(1) = 8, p < 0.005$. We only found a significant effect of *Technique* on *Mental*: $\chi^2_{Mental}(1) = 4.5$, $p < 0.05$.

All participants reported that it was hard to use Controller to select moving targets at a distance because it was difficult to estimate the depth of both targets and virtual controllers when in motion. The results above also indicated that *Position* increased both *Mental* and *Effort* loads. With the teleportation of the Controller, some participants felt that large body movement seemed to cause relatively smaller Controller movement due to perspective. Some also reported that visually smaller target width and hand tremor caused unexpected movement, especially when aiming at targets at a distance.

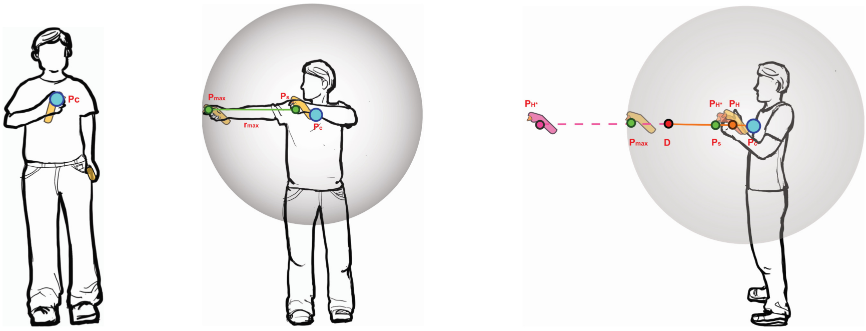
4.6 Discussion

When selecting moving targets, Controller is slower but more accurate than Ray-Casting. Figure 3 and user feedback reveal that *Position* impacts both Controller and Ray-Casting; more specifically, depth affects perceived visual size. For Ray-Casting, due to perspective, a distant moving target has visually smaller width, which causes difficulties for aiming. For Controller, in addition to targeting visually smaller targets, participants also must estimate depth, adding mental demand to selection, as shown in Fig. 4. *Speed* is a dominant factor affecting users' selection time and accuracy. Interpreting these graphical results in light of statistical analysis, these results argue that selection time and error rate is correlated to target speed.

5 Reach-Bound Go-Go

Examining the results above, we note that the virtual hand metaphor had significant accuracy advantage when selecting near targets (19.35%) versus at a distance (41.45%), and the teleportation of virtual hand resulted in high workload scores. This, then, leads to the question of whether we can enhance the virtual hand for more elegant distant targeting.

The technique we leverage to support enhanced distant targeting using the virtual hand metaphor is a variant of the *Go-Go* technique [43]. Earlier, we noted that the *Go-Go* technique leverages a non-linear mapping function to enable reaching beyond arms' length. The *Go-Go* technique maps virtual hand position to physical hand position up to a certain distance from the user. Beyond this range, the distance of the virtual hand is magnified by a multiplier (Fig. 5).



(a) (b) (c)

Fig. 5. rbGo-Go: (a)&(b) body posture calibration: P_C is recorded as the center position of a user's chest. r_{Max} is measured as the larger length between two arms' length. P_S is a shoulder position, and P_{Max} is the user's maximum reachable position. P_S and P_{Max} are recorded when a user stretch arms. (c) Motor space is divided into a linear mapping and non-linear mapping components by a tuned parameter D.

We call our modification to the *Go-Go* technique reach-bounded Go-Go interaction. Specifically, we restrict the movement space of (*Go-Go*), augment it with the body posture calibration as in [48], and simplify its configuration so that *rbGo-Go* can be used without the need for body tracking.

From [48], the amplified position of the virtual hand is defined as $P_{H^*} = P_C + f(r) * (P_H - P_C)$ where P_H is the physical hand position while holding a controller, P_C is the neural point which is defined as the center position of the chest. In *rbGo-Go*, as in [48], we use an amplification function, but the non-linear piece-wise amplification function $f(r)$ is a variant of Go-Go, taking the amplification slope and offset into consideration, as follows:

$$f(r) = \begin{cases} 1.0 & 0 \leq r \leq D \\ \left(\frac{L_{Max} - r_{Max}}{r_{Max} * (1-D)^2}\right) * (r - D)^2 + 1.0 & D < r \leq 1 \end{cases}$$

Here L_{Max} is the maximum length of the reachable space (e.g. bounded space in our experiment), r_{Max} is the maximum arm length, and D is the threshold that divides the physical and virtual hand mapping into direct mapped and non-linear parts. The value of D is $\frac{2}{3}$ based on the empirical experience from [43] but can be tuned in different scenarios. r is the physical offset defined as the ratio of the distance between P_H to P_C and P_{Max} to P_C: $r = \frac{|P_H - P_C|}{|P_{Max} - P_C|}$.

6 Follow-Up Study

Weaknesses in distant targeting for virtual hand resulted in *rbGo-Go*, but one open question is whether *rbGo-Go* enhances virtual hand interaction for moving targets. To test this, we increased the size of the bounded space (300 cm (depth) × 120 cm (height) × 120 cm (width)) to allow targets to move in a more general space and conducted a follow-up study evaluating *rbGo-Go* against Ray-Casting (Fig. 6).

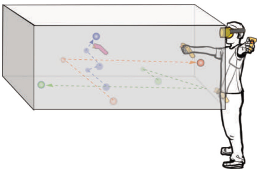


Fig. 6. The size-increased bounded space with possible target motion direction.

6.1 Experimental Design

A repeated-measure within-subject design was used. The independent variables (IVs) *Technique* (rbGo-Go, Ray-Casting), *Speed* (0.75 cm/frm, 1.00 cm/frm, and 1.25 cm/frm), *Width* (6 cm, 10 cm) and *Block* (1-4).

Each participant performed 4 blocks of trials and was instructed to select a static white sphere ("dummy" target) to start a block and control the initial position. Given the *Technique*, within each block, for each of the 3 target motion speeds in random order, 15 selections were made for each of the 2 target widths, also presented in random order. Six spheres, with only 1 blue sphere (goal target) were generated within the bounded space with the same *Speed* and same *Width* but moving in random direction. The experimental system moved to the next trial when the goal target was correctly selected. When a goal target was missed or a distractor was selected, the goal target did not disappear. Once correctly selected, the current goal target vanished and the next goal target would generate at a distance (40 cm, 80 cm, 120 cm in random order) from the virtual controller' position of rbGo-Go or hit position of Ray-Casting. When each participant finished a block, the static "dummy" target showed up and the participant could take breaks before selecting the "dummy" target. In summary, each participant performed 2 *Technique* × 3 *Speed* × 2 *Width* × 15 trials × 4 *Block* = 720 trials.

6.2 Procedure

The only difference in procedure for this study from our initial study was that participants were asked to calibrate the neutral point, shoulder position, and arms' length to generate the required parameters for *rbGo-Go*. During the experiment, participants were asked to limit changes in body posture to ensure stable chest position.

6.3 Participants

10 participants were recruited by word-of-mouth in our organization (ages 20 to 28 ($\mu = 23.8, \sigma = 2.7$), 6 male, 4 female, 1 left-handed), 3 experienced VR users. Given the Covid-19 pandemic, 6 participants from the initial study also took this study. A training session, including posture calibration and technique practice, took about 10 min, and the experiment lasted for 35 min.

6.4 Results

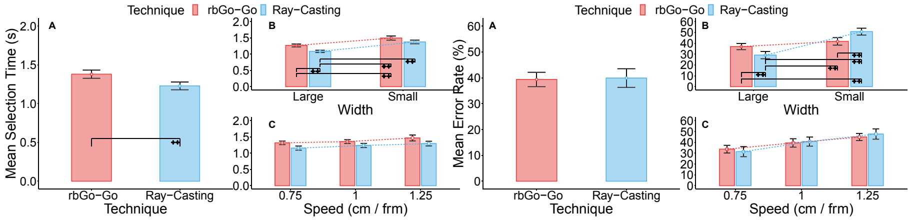


Fig. 7. Mean selection time (left) and mean error rate (right) for (A) *Technique*; (B) *Technique* by *Width*; (C) *Technique* by *Speed*. Error bars are shown with 95% confidence intervals. The statistic significances evaluated by pairwise t-test are marked with + (++ = p < 0.01 and + = p < 0.05).

Selection Time. After removing outliers (<1%), our data contained 7134 trials across 10 participants, a Box-Cox transformation ($\lambda = -0.38$) was applied to non-normal residuals of the selection time. We found a significant effect of *Block* ($F_{3,27} = 9.05$, $p < 0.001$, $\eta_p^2 = 0.11$). Pairwise t-test reported that Block 1 (mean = 1.44 s) took significantly longer ($p < 0.001$) than Block 3 (1.27 s) and Block 4 (1.27 s). Therefore, Block 1 was removed in the following analysis.

We found a significant effect of *Technique* ($F_{1,9} = 20.05$, $p < 0.005$, $\eta_p^2 = 0.13$) on the selection time. rbGo-Go (1.38 s) was significantly slower than Ray-Casting (1.23 s). We found a significant effect of *Speed* ($F_{1,9} = 23.22$, $p < 0.001$, $\eta_p^2 = 0.07$). Selecting targets moving at 0.75 cm/frm (1.24 s) was significantly faster than at 1.25 cm/frm (1.28 s, $p < 0.005$). We did not find a significant interaction effect between *Speed* and other IVs. We found a significant effect of *Width* ($F_{1,9} = 151.39$, $p < 0.001$, $\eta_p^2 = 0.29$). Selecting targets with large width (1.17 s) was significantly faster than small width (1.43 s, $p < 0.001$). We found a significant interaction effect between *Technique* and *Width* ($F_{1,9} = 8.35$, $p < 0.05$, $\eta_p^2 = 0.02$) on selection time. Considering large versus small widths, Ray-Casting's improved performance over rbGo-Go was primarily for large targets. While Ray-Casting was on average faster for both large and small targets, the corrected post-hoc difference was not statistically significant when considering only small targets.

Accuracy. In the absence of failing the normality assumption, we treat the residual of error rate as normal for analysis. Without Block 1, we did not find significance of *Block* on the error rate.

We did not find a significant effect of *Technique* on the error rate between rbGo-Go (mean = 39.34%) and Ray-Casting (39.90%). We found a significant effect of *Speed* ($F_{2,18} = 52.74$, $p < 0.001$, $\eta_p^2 = 0.24$). Selecting targets moving at 0.75 cm/frm (32.53%, $p < 0.001$) caused significantly less erroneous selection than the other two speeds. Also, selecting targets moving at 1.00 cm/frm (40.05%)

caused significantly less erroneous selection than at 1.25 cm/frm (46.28%, $p < 0.001$). We did not find any significant interaction effect between *Speed* and other IVs. We found a significant effect of *Width* ($F_{1,9} = 52.80$, $p < 0.001$, $\eta_p^2 = 0.30$). Selecting targets with large width (33.01%) caused significantly less erroneous selection than small width (46.23%, $p < 0.001$). We found a significant interaction effect between *Width* and *Technique* ($F_{1,9} = 99.97$, $p < 0.001$, $\eta_p^2 = 0.15$). Ray-Casting caused significantly higher error rate on small targets (50.72%) than large targets (29.07%, $p < 0.001$). However, rbGo-Go caused a similar error rate on targets with large width (36.94%) and small width (41.75%) (Fig. 8).

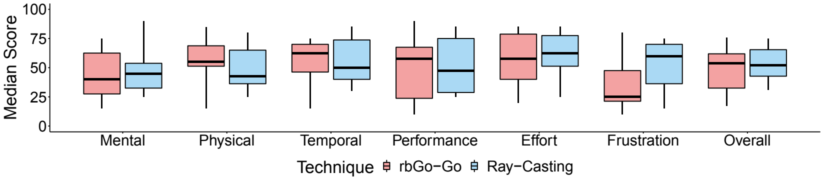


Fig. 8. Box plots for perceived task loads of the TLX questionnaire.

Task Loads Analysis and User Feedback. A Friedman test did not report any significance effect of *Technique* on any perceived task loads. In other words, rbGo-Go has similar loads for all attributes as Ray-Casting, while noticeably, rbGo-Go had lower much median score on *Frustration* than Ray-Casting.

Since rbGo-Go is a non-linear mapping between the real and virtual hands, participants found it hard to control this technique during early use, especially for targets moving at high speed. However, as they practiced, they (P0, P5, & P7-9) felt more confident and found it easier to select targets with small width and at distance, which was consistent with the learning effect we found in the selection time analysis. Target width was considered as an important factor affecting participants' performance (P6: *If not considering the technique, target width plays a very important role ... targets with small width at a distance are hard*). This was obvious for Ray-Casting, as several participants commented that it was hard to select targets with small width, and they perceived more hand jitter. Some participants (P6, P8) also reported that, using Ray-Casting, they found it harder to select targets moving up&down and left&right, compared with those moving towards and away from them, where rbGo-Go had the opposite feedback as depth estimation on targets was necessary.

6.5 Discussion

Similar to the former study, Ray-Casting was more efficient than rbGo-Go, but participants had a similar and high error rate. Graphically examining the results of Fig. 7, the selection time and error rate was positively related to *Speed* while negatively correlated to *Width*, as per Jagacinski et al.'s model [26] of moving target selection.

rbGo-Go, by design (virtual hand), has a larger selection area because the spatial extents of the hand are larger than those of a line (Ray-Casting). This contributed to improved accuracy for rbGo-Go despite difficulty controlling the non-linear mapping. On the other hand, Ray-Casting, or broadly virtual pointer, had a narrow ray such that selecting a small target became challenging. As participants noted, using Ray-Casting to select targets moving left&right and up&down was challenging because, to capture these targets, a narrow ray needed to translate a larger distance, during which a mis-selection could easily occur.

rbGo-Go simplifies the tuning process required by Go-Go. For rbGo-Go, the amplification parameter, k, which was manually adjusted in Go-Go, is now determined by the size of the interaction space, the arm length of a user, and a tuned threshold. It requires no additional hardware (e.g. a Kinect as in [48]) such that users can configure and execute the study remotely and independently. Though self-calibration by users may result in small variations in performance, it allowed us to better preserve social distancing requirements.

7 General Discussion

Considering both experiments, similar to results in 1D and 2D [22,26,32] targeting, *Speed* has a strong influence on the performance of virtual hand and virtual pointer metaphors. Specifically, higher motion speed causes longer selection time and higher error rate. Therefore, it is reasonable to design techniques that reduce motion speed, such as the Hook [3] and Target Ghost [21]. In terms of *Width*, interestingly, we observe an impact on these two metaphors in the moving target selection task similar to that observed in Poupyrev et al.'s static target selection study [42]: Compared with Ray-Casting, rbGo-Go is comparably fast and more accurate when selecting small objects. Additionally, compared with moving target selection tasks in 1D [24] and 2D [25], object depth will also impact users' perceived width. This problem is a speciality in 3D and more complicated when targets move in any direction. Since in our study, the impact of depth on target width is identical for both *Technique*s, we believe that this confound is controlled across conditions. Investigating this issue and exploring strategies to address it is another way to study these two metaphors.

One might be tempted to dismiss virtual hand as an interactive metaphor for moving target selection unless interaction is restricted to arms' length. After all, it is slower in both experiments, and only has an error rate advantage at close distances in the initial study. Furthermore, it is tempting to conclude that rbGo-Go serves no purpose due to its increased error rate compared to naive virtual hand. However, we would caution against such a simple interpretation.

First, virtual hand and virtual pointer are different pointing metaphors. Virtual hand asks a user to control all three, sweep, elevation, and depth, to fully target a unique, 3D location in space. Virtual pointer, in contrast, is an intersection-based technique where users can point at targets in a depth agnostic way. In many instances, virtual pointer is feasible. However, there may be moving target tasks where the goal location may not be identifiable by the system: as one example, imagine a virtual drawing application where a user selects an initial position and then draws a smooth trajectory along a desired path through the immersive environment. Interactions like this are not immediately possible via virtual pointer; they require some augmentation to control depth [7,19]. *rbGo-Go*, in contrast, can facilitate these tasks without enhancement.

Second, to allow virtual hand to target increased volumes, our *rbGo-Go* technique increases the range of movement of the virtual hand metaphor when the user reaches beyond a specific distance from themselves, D, but preserves behavior for distances less than D. For proximal targeting, users continue to benefit from the 1-to-1 direct mapping. As users reach beyond D, mapping smaller physical depth movements of the user onto larger depth transitions for the virtual hand metaphor means that, for *rbGo-Go*, targets are smaller in depth in motor space. The fact that error rate converged on Ray-Casting but with the added ability to specifically select depth supports the utility of a technique like rbGo-Go as an alternative to virtual pointer techniques, particularly in cases where depth must be controlled during targeting.

Recall that, compared with Ray-Casting, rbGo-Go is comparably fast and is more accurate when selecting small objects. Therefore, we would argue that rbGo-Go and Ray-Casting play off against each other in terms of advantages, i.e., that it is most important to understand the relative pros and cons of each metaphor in moving target acquisition. To summarize, there exist potential benefits to each metaphor in the context of moving target selection, as follows:

- Virtual hand: (1) Lower error rate for proximal moving target selection. (2) Higher immutability of error rate across different target widths. (3) Depth control.
- Virtual pointer: (1) Generally faster. (2) Consistent (though relatively high) error rate. (3) Existing rich facilitation techniques [7,14,18,35] for efficiency and stabilization.

Overall, we would argue that these advantages in moving target selection are useful data points for designers who wish to incorporate an ability to select moving targets into their virtual environment applications.

7.1 Future Work

Facilitation Techniques. One aspect we have not evaluated concerns facilitation techniques which could aid target acquisition. For virtual hand, depth cues (such as motion parallax [27], and visual guidance [7]) may simplify depth estimation of both targets and hands. For virtual pointer (Ray-Casting), increasing the effective size [18] and activation area [21] of targets, or the selection area of a ray (volume ray) [35] are promising approaches, although dynamic and elastic width caused by unpredictable motion may be a concern. Additionally, weakening the speed effect, e.g. transforming a dynamic selection task to a static selection task [3,21], and stabilizing control with filtering [14] are possible solutions to address hand jitter's and speed's impact on targeting, two common challenges for both metaphors.

Factors Beyond Speed and Width. Alongside target speed and width, there are other factors that could influence metaphor choices when considering the use of virtual hand and pointer metaphors in 3D VR/AR environments for the capture moving targets. As one example, the performance of these metaphors for crossing-based selection tasks [1], where users select targets by crossing a target's boundary instead of pointing inside its perimeter, is an open question. This crossing paradigm has its unique values as it can adapt to these two metaphors more naturally (e.g. avoiding the Heisenberg effect) and can also improve user performance in particular scenarios, for example the Saber Beat game. In addition, the use of these two metaphors in real-world VR/AR applications will raise questions about how various feedback techniques could affect users' immersive experience while selecting moving targets. For example, haptic feedback [12] on a virtual hand may enhance users' experience, and improved visual feedback on the ray during Ray-Casting [7] may improve users' environmental awareness.

8 Conclusion

Alongside our introduction of *rbGo-Go*, a variant of the *Go-Go* technique, we provide two empirical studies to compare virtual hand (Controller/*rbGo-Go*) and virtual pointer (Ray-Casting) metaphors in the context of moving target selection in virtual environments. Using a classic virtual hand metaphor (both proximal to the user and at distance), we find that virtual hand has a lower error rate in proximity to the user but slower selection time. Given the advantages and disadvantages of the basic virtual hand metaphor, we evaluate *rbGo-Go*, our modified version of the *Go-Go* technique. We find, again, that *rbGo-Go* is slower than Ray-Casting, but note advantages of the technique both in terms of small target precision and in terms of an ability to support target agnostic depth selection. We argue that the complementary advantages of the technique provide useful guidelines for designers of virtual environments when introducing interactions to support moving target acquisition.

Acknowledgements. We would like to thank Shino Che Yan for her help in creating Figs. 2, 5, and 6, Da-Yuan Huang for valuable feedback, participants for their help in this difficult time, and reviewers for valuable suggestions. This research received ethics clearance from the Office of Research Ethics, University of Waterloo. This research was funded by a grant from Waterloo-Huawei Joint Innovation Laboratory.

References

1. Accot, J., Zhai, S.: Beyond Fitts' law: models for trajectory-based HCI tasks. In: CHI 1997 Extended Abstracts on Human Factors in Computing Systems, CHI EA 1997, p. 250. Association for Computing Machinery, New York (1997). https://doi.org/10.1145/1120212.1120376
2. Accot, J., Zhai, S.: Refining Fitts' law models for bivariate pointing. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2003, pp. 193–200. Association for Computing Machinery, New York (2003). https://doi.org/10.1145/642611.642646
3. Al Hajri, A., Fels, S., Miller, G., Ilich, M.: Moving target selection in 2D graphical user interfaces, pp. 141–161 (2011). https://doi.org/10.1007/978-3-642-23771-3_12
4. Argelaguet, F., Andujar, C.: Efficient 3D pointing selection in cluttered virtual environments. IEEE Comput. Graphics Appl. **29**(6), 34–43 (2009)
5. Argelaguet, F., Andujar, C.: A survey of 3D object selection techniques for virtual environments. Comput. Graph. **37**(3), 121–136 (2013)
6. Arora, R., Kazi, R.H., Kaufman, D.M., Li, W., Singh, K.: Magicalhands: mid-air hand gestures for animating in VR. In: Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology, UIST 2019, pp. 463–477. Association for Computing Machinery, New York (2019). https://doi.org/10.1145/3332165.3347942
7. Baloup, M., Pietrzak, T., Casiez, G.: Raycursor: a 3D pointing facilitation technique based on raycasting. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019, pp. 1–12. Association for Computing Machinery, New York (2019). https://doi.org/10.1145/3290605.3300331
8. Baudisch, P., et al.: Drag-and-pop and drag-and-pick: techniques for accessing remote screen content on touch- and pen-operated systems (2003)
9. Bi, X., Li, Y., Zhai, S.: Ffitts law: modeling finger touch with Fitts' law. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2013, pp. 1363–1372. Association for Computing Machinery, New York (2013). https://doi.org/10.1145/2470654.2466180
10. Bideau, B., Kulpa, R., Vignais, N., Brault, S., Multon, F., Craig, C.: Using virtual reality to analyze sports performance. IEEE Comput. Graphics Appl. **30**, 14–21 (2010). https://doi.org/10.1109/MCG.2009.134
11. Bowman, D.A., Hodges, L.F.: An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In: Proceedings of the 1997 Symposium on Interactive 3D Graphics, pp. 35–ff (1997)
12. Buchmann, V., Violich, S., Billinghurst, M., Cockburn, A.: Fingartips: gesture based direct manipulation in augmented reality. In: Proceedings of the 2nd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia, GRAPHITE 2004, pp. 212–221. Association for Computing Machinery, New York (2004). https://doi.org/10.1145/988834.988871
13. Cashion, J., Wingrave, C., Laviola, J.J.: Dense and dynamic 3D selection for game-based virtual environments. IEEE Trans. Visual. Comput. Graphics 634–642 (2012)
14. Casiez, G., Roussel, N., Vogel, D.: 1€filter: a simple speed-based low-pass filter for noisy input in interactive systems. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2012, pp. 2527–2530. Association for Computing Machinery, New York (2012). https://doi.org/10.1145/2207676.2208639

15. Chen, Y., Katsuragawa, K., Lank, E.: Understanding viewport- and world-based pointing with everyday smart devices in immersive augmented reality. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, CHI 2020, pp. 1–13. Association for Computing Machinery, New York (2020). https://doi.org/10.1145/3313831.3376592
16. Debarba, H.G., Grandi, J.G., Maciel, A., Nedel, L., Boulic, R.: Disambiguation canvas: a precise selection technique for virtual environments. In: Kotzé, P., Marsden, G., Lindgaard, G., Wesson, J., Winckler, M. (eds.) INTERACT 2013. LNCS, vol. 8119, pp. 388–405. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40477-1_24
17. Fitts, P.M.: The information capacity of the human motor system in controlling the amplitude of movement. J. Exp. Psychol. **47**(6), 381–91 (1954)
18. Grossman, T., Balakrishnan, R.: The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2005, pp. 281–290. Association for Computing Machinery, New York (2005). https://doi.org/10.1145/1054972.1055012
19. Grossman, T., Balakrishnan, R.: 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, UIST 2006, pp. 3–12. Association for Computing Machinery, New York (2006). https://doi.org/10.1145/1166253.1166257
20. Hart, S.G.: Nasa-task load index (NASA-TLX); 20 years later. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 50, no. 9, pp. 904–908 (2006). https://doi.org/10.1177/154193120605000909
21. Hasan, K., Grossman, T., Irani, P.: Comet and target ghost: techniques for selecting moving targets. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2011, pp. 839–848. Association for Computing Machinery, New York (2011). https://doi.org/10.1145/1978942.1979065
22. Hoffmann, E.R.: Capture of moving targets: a modification of Fitts' law. Ergonomics **34**(2), 211–220 (1991). https://doi.org/10.1080/00140139108967307
23. Huang, J., et al.: Modeling the endpoint uncertainty in crossing-based moving target selection. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, CHI 2020, pp. 1–12. Association for Computing Machinery, New York (2020). https://doi.org/10.1145/3313831.3376336
24. Huang, J., Tian, F., Fan, X., Zhang, X.L., Zhai, S.: Understanding the uncertainty in 1D unidirectional moving target selection. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI 2018, pp. 1–12. Association for Computing Machinery, New York (2018). https://doi.org/10.1145/3173574.3173811
25. Huang, J., Tian, F., Li, N., Fan, X.: Modeling the uncertainty in 2D moving target selection. In: Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology, UIST 2019, pp. 1031–1043. Association for Computing Machinery, New York (2019). https://doi.org/10.1145/3332165.3347880
26. Jagacinski, R.J., Repperger, D.W., Ward, S.L., Moran, M.S.: A test of Fitts' law with moving targets. Hum. Factors **22**(2), 225–233 (1980)
27. Jones, A., Swan, J.E., Singh, G., Kolstad, E.: The effects of virtual reality, augmented reality, and motion parallax on egocentric depth perception. In: 2008 IEEE Virtual Reality Conference, pp. 267–268 (2008)

28. Kabbash, P., Buxton, W.A.S.: The "prince" technique: Fitts' law and selection using area cursors. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 1995, pp. 273–279. ACM Press/Addison-Wesley Publishing Co., USA (1995). https://doi.org/10.1145/223904.223939
29. Khamis, M., Oechsner, C., Alt, F., Bulling, A.: Vrpursuits: interaction in virtual reality using smooth pursuit eye movements. In: Proceedings of the 2018 International Conference on Advanced Visual Interfaces, AVI 2018. Association for Computing Machinery, New York (2018). https://doi.org/10.1145/3206505.3206522
30. Klopcar, N., Lenarcic, J.: Kinematic model for determination of human arm reachable workspace. Meccanica **40**, 203–219 (2005)
31. Kopper, R., Bacim, F., Bowman, D.A.: Rapid and accurate 3D selection by progressive refinement. In: 2011 IEEE Symposium on 3D User Interfaces (3DUI), pp. 67–74 (2011)
32. Lee, B., Oulasvirta, A.: Modelling error rates in temporal pointing. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, CHI 2016, pp. 1857–1868. Association for Computing Machinery, New York (2016). https://doi.org/10.1145/2858036.2858143
33. Li, J., Cho, I., Wartell, Z.: Evaluation of cursor offset on 3D selection in VR. In: Proceedings of the Symposium on Spatial User Interaction, pp. 120–129. Association for Computing Machinery, New York (2018). https://doi.org/10.1145/3267782.3267797
34. Li, Y., Wu, D., Huang, J., Tian, F., Wang, H., Dai, G.: Influence of multi-modality on moving target selection in virtual reality. Virtual Reality Intell. Hardware **1**(3), 303–315 (2019). https://doi.org/10.3724/SP.J.2096-5796.2019.0013
35. Lu, Y., Yu, C., Shi, Y.: Investigating bubble mechanism for ray-casting to improve 3D target acquisition in virtual reality. In: 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), pp. 35–43. IEEE Computer Society, Los Alamitos (2020). https://doi.org/10.1109/VR46266.2020.00021
36. MacKenzie, I.S.: Fitts' law as a research and design tool in human-computer interaction. Hum.-Comput. Interact. **7**(1), 91–139 (1992). https://doi.org/10.1207/s15327051hci0701_3
37. McGuffin, M., Balakrishnan, R.: Acquisition of expanding targets. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2002, pp. 57–64. Association for Computing Machinery, New York (2002). https://doi.org/10.1145/503376.503388
38. Mine, M., Brooks, F., Jr., Sequin, C.: Moving objects in space: exploiting proprioception in virtual-environment interaction (1997). https://doi.org/10.1145/258734.258747
39. Olwal, A., Feiner, S.: The flexible pointer: an interaction technique for selection in augmented and virtual reality (2003)
40. Ortega, M.: Hook: heuristics for selecting 3D moving objects in dense target environments. In: 2013 IEEE Symposium on 3D User Interfaces (3DUI). IEEE, March 2013. https://doi.org/10.1109/3dui.2013.6550208
41. Pheasant, S., Haslegrave, C.: Bodyspace: Anthropometry, Ergonomics and the Design of Work (2018). https://doi.org/10.1201/9781315375212
42. Poupyrev, I., Ichikawa, T., Weghorst, S., Billinghurst, M.: Egocentric object manipulation in virtual environments: empirical evaluation of interaction techniques. In: Computer Graphics Forum, vol. 17, no. 3, pp. 41–52 (1998). https://doi.org/10.1111/1467-8659.00252

43. Poupyrev, I., Billinghurst, M., Weghorst, S., Ichikawa, T.: 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, UIST 1996, pp. 79–80. Association for Computing Machinery, New York (1996). https://doi.org/10.1145/237091.237102
44. Steinicke, F., Ropinski, T., Hinrichs, K.: Object selection in virtual environments using an improved virtual pointer metaphor. In: Wojciechowski, K., Smolka, B., Palus, H., Kozera, R., Skarbek, W., Noakes, L. (eds.) Computer Vision and Graphics, pp. 320–326. Springer, Dordrecht (2006). https://doi.org/10.1007/1-4020-4179-9_46
45. Su, X., Au, O.K.C., Lau, R.W.: The implicit fan cursor: a velocity dependent area cursor. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2014, pp. 753–762. Association for Computing Machinery, New York (2014). https://doi.org/10.1145/2556288.2557095
46. Sun, J., Stuerzlinger, W.: Extended sliding in virtual reality. In: 25th ACM Symposium on Virtual Reality Software and Technology, VRST 2019. Association for Computing Machinery, New York (2019). https://doi.org/10.1145/3359996.3364251
47. Tresilian, J.: Hitting a moving target: perception and action in the timing of rapid interceptions. Percept. Psychophys. **67**, 129–149 (2005). https://doi.org/10.3758/BF03195017
48. Wentzel, J., d'Eon, G., Vogel, D.: Improving virtual reality ergonomics through reach-bounded non-linear input amplification. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, CHI 2020, pp. 1–12. Association for Computing Machinery, New York (2020). https://doi.org/10.1145/3313831.3376687
49. Williams, E.: Experimental designs balanced for the estimation of residual effects of treatments (1949). https://doi.org/10.1071/CH9490149
50. Wobbrock, J.O., Cutrell, E., Harada, S., MacKenzie, I.S.: An error model for pointing based on Fitts' law. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2008, pp. 1613–1622. Association for Computing Machinery, New York (2008). https://doi.org/10.1145/1357054.1357306
51. Wobbrock, J.O., Jansen, A., Shinohara, K.: Modeling and predicting pointing errors in two dimensions. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2011, pp. 1653–1656. Association for Computing Machinery, New York (2011). https://doi.org/10.1145/1978942.1979183
52. Wyss, H.P., Blach, R., Bues, M.: isith - intersection-based spatial interaction for two hands. In: 3D User Interfaces (3DUI 2006), pp. 59–61 (2006)
53. Yoon, J.W., Jang, S.H., Cho, S.B.: Enhanced user immersive experience with a virtual reality based FPS game interface. In: Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games, pp. 69–74 (2010). https://doi.org/10.1109/ITW.2010.5593369
54. Yu, D., Liang, H.N., Lu, X., Fan, K., Ens, B.: Modeling endpoint distribution of pointing selection tasks in virtual reality environments. ACM Trans. Graph. **38**(6) (2019). https://doi.org/10.1145/3355089.3356544
55. Zhai, S., Kong, J., Ren, X.: Speed–accuracy tradeoff in Fitts' law tasks–on the equivalency of actual and nominal pointing precision. Int. J. Hum.-Comput. Stud. **61**(6), 823–856 (2004). https://doi.org/10.1016/j.ijhcs.2004.09.007. Fitts' law 50 years later: applications and contributions from human-computer interaction