Towards Better Throwing: A Comparison of Performance and Preferences Across Point of Release Mechanics in Virtual Reality

Amirpouya Ghasemaghaei
aghaei.ap@ucf.edu
University of Central Florida
Orlando, Forida, USA

Mykola Maslych
maslychm@gmail.com
University of Central Florida
Orlando, Forida, USA

Yahya Hmaiti
yohan.hmaiti@ucf.edu
University of Central Florida
Orlando, Forida, USA

Esteban Segarra Martinez
esteban.segarra@ucf.edu
University of Central Florida
Orlando, Forida, USA

Joseph J. LaViola Jr.
jlaviola@ucf.edu
University of Central Florida
Orlando, Forida, USA

ABSTRACT

An underexplored interaction metaphor in virtual reality (VR) is throwing, with a considerable challenge in achieving accurate and natural results. We conducted an empirical investigation of participants' performance in a VR throwing task, measuring their accuracy and preferences across Point of Release (PoR) mechanics (manual and automatic) with various input device categories (hand-held, on-body, external) and throwable object types. Participants were tasked with throwing a baseball, a bowling ball, and a football toward targets using 5 input configurations (2 manual and 3 automatic PoR). Results from 30 participants indicate that the overall highest accuracy was achieved with an automatic PoR configuration (on-body tracker). The post-study and VR survey results indicate that the majority of participants preferred a manual PoR configuration (hand-held VR controller-derived) for the throwing direction, throwing speed, and as being the closest to real-life throwing. Our findings are useful for VR researchers and developers who want to implement throwing as a technique in their applications.

CCS CONCEPTS

• **Human-centered computing → Virtual reality**; **Usability testing**; *Empirical studies in HCI*; Gestural input.

KEYWORDS

Virtual reality, throwing, PoR, input devices, interaction

ACM Reference Format:
Amirpouya Ghasemaghaei, Mykola Maslych, Yahya Hmaiti, Esteban Segarra Martinez, and Joseph J. LaViola Jr.. 2024. Towards Better Throwing: A Comparison of Performance and Preferences Across Point of Release Mechanics in Virtual Reality. In *Graphics Interface (GI '24), June 03–06, 2024, Halifax, NS, Canada.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3670947.3670963

1 INTRODUCTION

Virtual Reality (VR) enables its users to be immersed in virtual environments (VEs) where they can interact with different VE components as they would in real-life [18]. Currently, interaction paradigms in VR mostly rely on the use of handheld controllers to perform different actions, including object selection, locomotion, and manipulation [18]. Virtual object throwing is an interaction that can have a very different user experience to its real-world counterpart [37]. This difference is also expressed by the large community of VR users [1, 6, 13, 16, 21, 22, 25]. Throwing is a common interaction element in real-world sports and games and motivates natural and intuitive throwing dynamics for VE scenarios in VR. Through more intuitive experiences, users can confidently rely on throwing as a mechanic for completing tasks in VR.

Prior work on this topic explored VR throwing and its applications. Improvements in throwing realness and accuracy include incorporating aerodynamics, controller-based velocity calculation, usage of Tangible User Interfaces (TUIs), among others [5, 29, 37]. Throwing was used across different applications, including sports training, rehabilitation, and motor skills improvement [10, 19, 26]. Recent research advances permitted gesture-based interactions through unmediated interactions without the need to continuously carry hand-held devices [18], which paves the way for more intuitive and authentic immersive experiences. This diversity in input devices and body tracking mechanisms can potentially improve throwing dynamics in VR. Using different input device categories, our research investigates whether throwing performance differs across different Point of Release (PoR) mechanics, where PoR is the exact time (and/or location) at which a virtual object is detached from its root object and begins following its trajectory based on its initial velocity and direction (see subsection 3.1 and Figure 1). We compared the capabilities of five input configurations with varying sensing capabilities and throwing mechanisms. In this process, we answered the following research questions: **(RQ.1):** *Which throwing configurations are most/least preferred based on participant perception?* **(RQ.2):** *While throwing, which throwing configuration felt more realistic based on participant perception?* **(RQ.3):** *While throwing, which throwing configuration was perceived to be more accurate based on participant perception?* **(RQ.4):** *In what type of throwing (underarm / overarm) would participants perform better?* **(RQ.5):** *What are the differences between objective and participant perceived throwing performance?*

GI '24, June 03–06, 2024, Halifax, NS, Canada

ACM ISBN 979-8-4007-1828-1/24/06
https://doi.org/10.1145/3670947.3670963

We conducted a 5 × 3 within-subject study, where we varied the PoR mechanics using different input device categories and throwable object types. The hand-held category included two manual and one automatic PoR throwing configuration (controller-based), the on-body and external categories each had one automatic PoR (threshold-based) throwing configuration (Vive tracker and Kinect sensor). The throwable objects (baseball, football, bowling) were designed for overarm and underarm throwing. Participants performed better in underarm throwing (bowling), and overall, the highest accuracy was achieved with the on-body tracker (Vive tracker). The collected survey data revealed that a hand-held throwing configuration (controller press) was perceived as most accurate, while another hand-held throwing configuration (controller hold) was most favored. These results contrast the objective accuracy metrics for the best device because participants preferred a more commonly used input device category that offers more control over the PoR.

2 RELATED WORK

Prior work on VR throwing covered topics in tracking accuracy [27, 29, 34, 37], sports and rehabilitation training [7, 8, 10, 19, 26], PoR prediction [30, 33], motor abilities [10, 14, 23, 24], embodiment [2–4], haptics [15, 28], and differences between real-life and VR throwing [5, 36, 37]. Our study mainly focused on comparing multiple tracking devices using different PoR. To implement this, we relied on insights from the work summarized below.

Borgwardet et al. [4] compared throwing a virtual Frisbee with and without a hand representation enabled. They found that enabling a hand representation improved the accuracy of throwing. In our study, we decided not to include hand representation to keep the overall experience consistent across devices. Winkler et al. [29] compared the weight perception of virtual throwable objects when aerodynamic simulation was enabled and disabled. They found that the weight of light objects was perceived more accurately where simulated aerodynamics were included. Yamac et al. [30] developed a model that predicts throwing release points from a dataset of throwing motions collected with a Vicon motion capture system. They reported that all users felt that overarm throws were less realistic than underarm throws because of the force and effort needed to throw the ball. Singh et al. [23] explored participants' VR throwing performance in a Dual-Motor-Task (motor and cognitive skills). Participants performed walking on a treadmill, throwing in VR while stationary, and throwing in VR while walking on a treadmill. The results revealed a correlation between hit point localization variation and cognitive ability and the effect of motor ability on the choice of orienting the throwing plane in space. Bozgeyikli et al. [5] used two Vive trackers inside a transparent physical ball (Tangiball) to explore TUIs in VR. They compared Tangiball to a virtual-only ball where participants had to kick it onto a virtual target on the ground, using Vive trackers to track each foot. The authors reported that the TUI improved participants' performance and task realness. Zindulka et al. [37] compared in-VR and real-world throwing for three throwing styles: overhand, underhand, and overhand at a greater distance. The authors reported that throwing in VR is less accurate than in real-life because of potential differences in the required effort, kinematic patterns, and difficulties in timing the release. The authors suggested designing larger targets until throwing mechanics in VR improve.

In the summarized research, some used the Meta Quest [4, 29, 30], while others used the HTC Vive [5, 23, 37]. For hand tracking, VR controllers were the most commonly used [4, 23, 29, 37], however, some used motion capture [30, 32, 37] or a Vive tracker [5]. Most prior work used concentric circles as targets and sphere-shaped balls as throwable objects. In all research focusing on realism, release velocity and direction of the throwable objects were set based on VR controller velocity over some number of past frames (or hand, in the absence of a controller). As for metrics, scoring based on the ring of the target [5, 23], or distance to the center [4, 37] were commonly used, and in some studies these results were shown to the participants on a scoreboard while the trials were ongoing [4, 5, 23]. Our study is grounded on some prior work; however, unlike what was done before, we used different PoR mechanics with multiple devices across the throwing tasks. The target design, throwable shapes, and setting of the release velocity are also following prior work. We collected and compared actual and perceived participant performance and preferences alongside post-study survey data. We did not use extra scoreboards that show past accuracy but only relied on visually indicating the object's landing location relative to the target.

3 METHODOLOGY

3.1 Point of Release

To simulate realistic throwing in VR, we need to approximate real-life behaviors of objects as closely as possible. This requires accurately setting the initial physical state of an object (velocity and direction) and simulating environmental factors (gravity, air resistance, and collisions) through a physics engine. We were less concerned with environmental factors and instead focused more on setting the objects' initial physical state. As defined in the Introduction section, PoR refers to the exact time/location at which a virtual object is detached from its root and begins following its trajectory based on its initial velocity and direction. Thus, accurate PoR, initial velocity, and direction lead to realistic-looking throwing in VR. We expand on setting velocity and environmental factors in subsection 3.5.

Throwing PoR can be specified either manually or automatically. Manual PoR is user-dependent since it is the user who defines where to release the throwable, whereas automatic PoR is motion feature-dependent (see Figure 1) [30, 31]. The most realistic but not very practical way to simulate a thrown object in VR is to throw a physical object that is tracked and rendered as part of the VE. This is achieved with either inside-out (IMUs, cameras) or outside-in (cameras, motion capture devices) sensors. These sensors send real-time information about the thrown object's position to the simulation engine [7]. Using this approach, the user has manual control over the PoR since the virtual thrown object will be released from the virtual hand simultaneously as the physical object is released from a real hand [4, 5, 7, 30]. While accurate, this method requires throwing an actual object out-of-VR, and our goal was to have an in VR-experiment. Manual PoR detection can also be achieved by

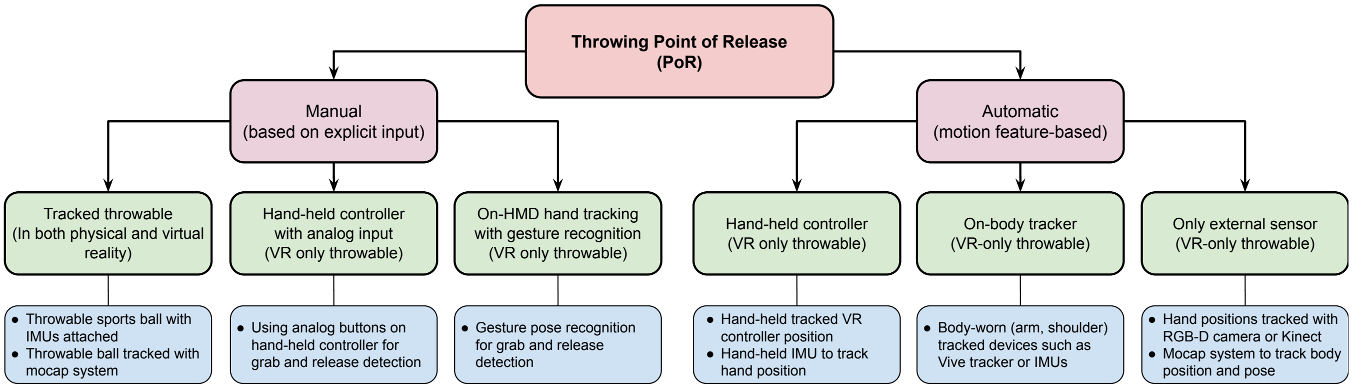

Figure 1: Possible ways to detect point of release (PoR) for throwing in VR with example sensing approaches. Manual PoR detection requires the actual throwing of an object or indicating release with analog buttons or gesture poses. Automatic PoR detection takes only positional information to assume the PoR based on motion features.

using a handheld controller with analog input, where the participant presses or un-presses a button to select or throw. Another possible way for manual PoR detection would be to use on-HMD sensors to track the participants' hands and use gesture recognition for detecting the grab and release gesture poses, however, this method is highly dependent on the FOV of the on-HMD sensor as tracking is lost when the participants' hands leave the tracked area. Automatic approaches for detecting the PoR are based on motion features extracted from hand-held controllers, on-body trackers, or external sensors [3, 32]. We utilized multiple devices with various sensing capabilities to cover the three main input device categories (hand-held, on-body, and external) and implemented some of the introduced tracking methods, called throwing configurations. They relied on different methods for PoR detection, and we compared them in a VR throwing task.

3.2 Throwing Configurations

We used the following devices to implement the PoR mechanics: HTC Vive controller, HTC Vive tracker, and Microsoft Kinect. These devices are off-the-shelf, popular, and offer three distinct tracking approaches (hand-held, on-body, and external). All these devices must integrate with the same HMD. We are aware of the presence of devices such as RealSense or UltraLeap. However, we decided to use Kinect for the external throwing configuration as it is more suitable specifically for full-body tracking (see section 6). For the Vive controller, we implemented 1 automatic and 2 manual PoR mechanics. We only used automatic PoR mechanics for the other devices (see Figure 2). The resulting throwing configurations cover 4/6 mechanics presented in Figure 1. Note that due to the nature of the manual approaches (CH, subsubsection 3.2.2 and CP, subsubsection 3.2.3), participants first had to select the throwable object using analog input, while for the automatic approaches, the object was already attached to participants' hand. We ensured that the selection step did not confound the study by having the participants practice each condition multiple times (see subsection 3.9), focusing on the throwing step so that the actual throwing evaluation was the same across conditions.

3.2.1 Vive on-body tracker (VT). This tracker was attached to the participant's dominant wrist using a strap. At the start of the trial, the throwable object was already visible and attached to where the participant's hand would be in the VE. This configuration used a thresholding-based method (automatic PoR) (see subsection 3.3). In the beginning, to define the threshold origin participants had to stand still, and based on the throwable type, they had to either put their hand next to their hip (bowling) or next to their head (football and baseball). Afterward, participants informed the researcher when they were ready to throw, and upon receiving confirmation, the researcher activated the throwing by enabling the threshold using a keyboard button and informed the participants that they could throw. Participants could move freely to perform the throw, and the throwable was released once their hand crossed the threshold distance from the origin.

3.2.2 Controller Hold (CH). Participants held the VR controller in their dominant hand, when using this controller, the throwable object was on the ground before them. This controller used a red ray to indicate the active input configuration. The participants selected the object by pointing the ray at it and pressing the trigger button to confirm the selection. Participants had to hold the trigger button to keep the throwable in hand; they could then freely move it and throw it by releasing the trigger (manual PoR).

3.2.3 Controller Press (CP). This resembled *controller hold*, but used a purple ray instead of a red ray as a visual cue of the active input configuration. Participants pointed to the throwable object and pressed the trigger to select it. Unlike *controller hold*, they did not need to hold the trigger continuously to keep the object in their hand. When performing the throwing gesture, they pressed the trigger again during the final phase of throwing to release the object.

3.2.4 Controller Threshold (CT). This method differed from other controller-based throwing configurations since the throwable was visible and directly attached to the controller at the trial's start, eliminating the need for point selection from the ground. The throwing mechanic resembled the VT with different threshold distances.

3.2.5 Microsoft Kinect V2.0 (K). We included a full-body tracking-based input device. We used a Kinect sensor placed on a table located 3m in front of the participant's start location and at a height of 1.8m above the ground. The received data from Kinect was noisy, so we used double-exponential smoothing [17] when applying joint positions to the participants' VR representation. We set the smoothing parameters to ($\alpha = 0.3, \beta = 0.5$), which visibly reduced the jitter of throwable objects while keeping latency not noticeable. To prevent device-related malfunctions, we reset the Kinect sensor for each participant before the start of the task. This cleared the body skeleton positions that Kinect holds in its memory. For this configuration, the throwable was also visible and already attached to the participants' tracked hand inside the VE. Given that the coordinate system origins of the Vive HMD and Kinect were different, to standardize the position of the palm joint, the system computed the offset between the participant's head and hand joints in Kinect's coordinates, and then added this offset to the HMD position each frame. The throwing dynamic for this configuration was similar to the VT but with a different threshold distance value.

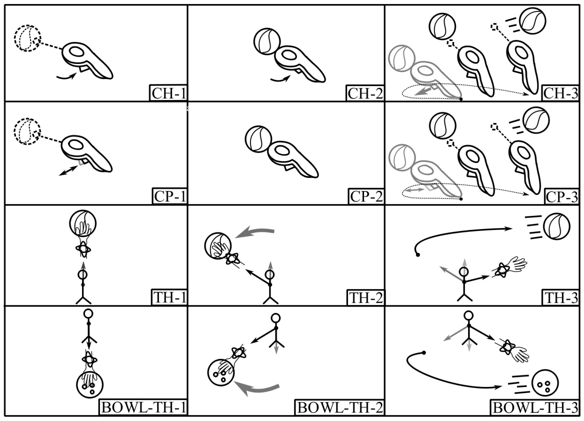

Figure 2: The overall procedure of throwing configurations. (CH): (1) Point at the object, hold down the trigger, (2) Keep holding, (3) Release to throw, (CP): (1) Point at the object and press the trigger, (2) No need to hold, (3) Press again to throw, (TH): (1) Hold the hand next to head for origin in overarm throwing, (2) Aim and start throwing gesture after confirmation, (3) Throw by going pass the threshold, (BOWL-TH): (1) Hold the hand next to the hip for origin in underarm throwing, (2) Aim and start throwing gesture after confirmation, (3) Throw by going pass the threshold.

3.3 Threshold Personalization

Our PoR method must work across all our input devices because hand-held and on-body devices only track a single joint (palm for Kinect, wrist for VT, and the actual controller for CT). Distance-based thresholding fit the requirements, so we proceeded with it. At an early stage of implementation we noticed that overarm throwing is mostly done when the hand is almost straight (past the highest point of the arm's arc), and underarm throwing is mostly done when the hand is in front of the body and is slightly bent. This means that threshold distance is affected by the arm length. We used one of the authors as a model and determined a set of threshold values that worked best for each device and throwable object (Table 1). We recorded the distance from the author's hand-held controller (fully extended down) to the HMD to serve as a reference scale (0.94m). During additional pilots with new participants, we recorded the HMD to hand distance for each participant and multiplied the calibration threshold values by the ratio of the newly recorded distance to the reference scale. Pilot participants confirmed that throwing matched their expected outcome, validating our threshold setting method. Later in the actual user study, the threshold distances for release points were computed as above. Dynamically adapting the threshold to participants with diverse arm spans enabled a degree of personalization.

Table 1: Threshold distances for the reference participant with a 0.94 meter distance from fully extended down hand to HMD. ***To use these values, record the mentioned distance, divide it by 0.94, and multiply it by the table values.***

Device	Bowling	Baseball	Football
CT	.38	.25	.25
VT	.40	.30	.35
K	.40	.30	.25

3.4 Throwable Object, Target, and Environment Design

Our VR experiment was designed using Unity3D 2021.3.27f1. We used three dissimilar throwable objects for overarm and underarm throwing (baseball, football, and bowling ball) (see Figure 3-(d)). During the throwing, we applied air friction and physics-based throwing dynamics so that the balls' behavior after release was as close to reality as possible. We describe the formulas and adjustments in detail in subsection 3.5. We also ensured that the position of the virtual throwable object matched the position of a similar physical object if it was held.

The user study VE was a simple football stadium used for both training and the actual experiment. Each condition had 6 identical targets at different positions, presented in random order, one per trial. The targets had 4 circular layers (see Figure 3) displayed as follows: a red central layer (Bullseye)($r = 0.35m$), a yellow layer ($r = 0.7m$), a green layer ($r = 1.05m$), and a final blue layer ($r = 1.4m$), the overall distance from the bullseye to the edge of the target was 1.4m. For bowling, targets were flat on the ground (see Figure 3-(b)). For baseball and football, targets were located mid-air (see Figure 3-(a)). In total, we had six mid-air and six on-ground targets, for both categories, two targets were to the participant's left, two in front, and two to their right. For each direction, one target was at 5m, and one was at 10m away from the participant.

3.5 Throwing Dynamics and Metaphors Design

When implementing the throwing metaphors, we aimed to make the throwing action as real as possible, and with that in mind, we used the velocity of the participant's hand movement acquired from the input configuration. This method relied on a first-order backward finite-difference method on the position calculated over a window of frames to save the current hand position subtracted by

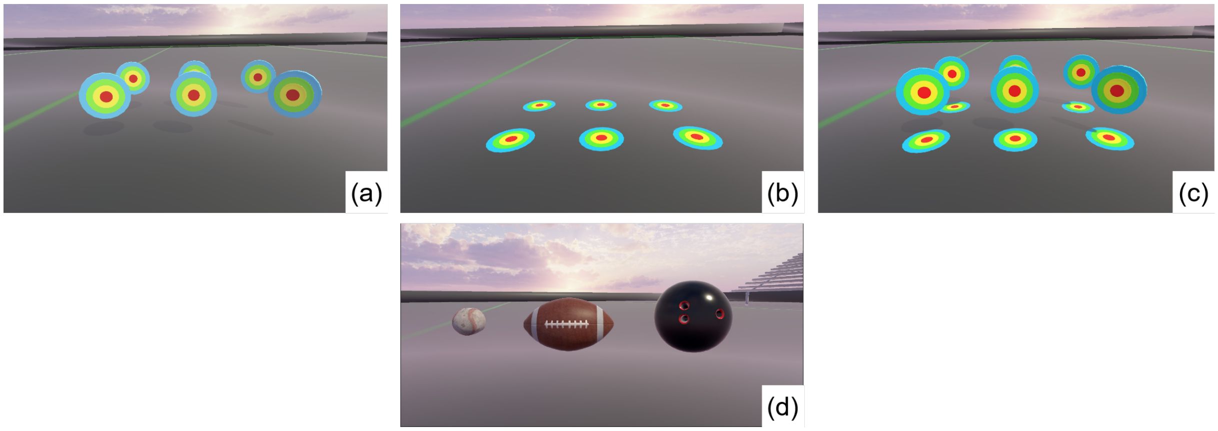

Figure 3: All the targets and throwables used in our experiment from an upper view perspective. (a) All mid-air targets shown at once, (b) All on-ground targets shown at once, (c) All mid-air and on-ground targets combined, (d) All throwable objects used in our study.

its position in the last frame for a constant number of frames (i.e., five or ten frames based on the device) [30, 37]. When the throwing was activated, the velocity of the participant’s hand was calculated by summing those saved values and dividing them by the timelapse. The resulting velocity vector was then applied to the throwable. The calculation followed a formula

$$V = \frac{\sum_{i=1}^{n}(P_i - P_{i-1})}{\sum_{i=1}^{n}\Delta T_i}, \tag{1}$$

where V is the velocity vector, n is the constant number of frames, P is the position of the participant’s hand at frame i, and T is the duration of the frame during which P was collected. We defined different frame window lengths for each input configuration, taking into account each device’s sampling rate by converting frames to real time through unity’s delta time function[1]. The window length was 5 frames for CH, CP and CT, and 10 frames for VT and K.

As Dunn et al. [11] stated, implementing physics in Unity is just an estimation from the real world. We tried to make the throwable object’s movement and curvature after throwing realistic by conducting numerous pilot tests with different participants, using gravity and air-resistance by modeling the drag force, then casting rays in the opposite direction to the throwable’s movement, and applying that force at the points where the rays hit the object[2]. We tried to make the throwable objects mimic the characteristics of their real counterparts post impact, so in our design, the baseball and football would bounce, and the bowling ball would slide. Also, to make the throwing realistic and avoid adding a confounding variable, we omitted any type of throwing trajectory indications showing the projected path of throwable after release.

3.6 Study Design and Variables

Our experiment was a within-subjects study with 2 independent variables (*input configuration*, and *throwable object type*). For the input configuration, we had 5 levels (controller hold, controller press, controller threshold, Vive tracker, and Kinect). For the throwable object type, we had 3 levels (bowling, baseball, and football). Using a within-subject design ensured consistency in results as it minimized the effects of the discrepancy between the throwing abilities of participants. We described the design of the VE and its components in subsection 3.4. We note that although there were 6 targets with different locations, they were not included as a factor in our experiment, considering that our main focus was on the throwing input configuration and the nature of the throwable only. In total, we had 15 condition combinations. The conditions were presented to the participants in order set by a counterbalanced Latin square. For each condition combination, we had 6 trials, and for each trial, the sequences of 6 targets were randomized. In our experiment, we recorded the throwing accuracy as the minimum Euclidean distance (in meters) of the projectile from the bullseye of the circular target. Participants were prompted to answer a VR questionnaire after completing each set of trials per study condition, and at the end of the experiment, they were asked to fill out a survey assessing their preferences (see subsection 3.9).

3.7 Apparatus

To avoid adding the HMD as a variable in our study, all input devices must be compatible with the same HMD. Initially, we tried Oculus Quest 2 as an egocentric tracking device, however, its controller’s position tracking is lost whenever the participant’s hand goes out of the hand and controller tracking range (a common gesture for overarm and underarm throwing). The Vive tracker also does not have a direct integration with Oculus Quest 2. Thus, we used HTC Vive HMD because it is widely used, available off-the-shelf, and because it uses lighthouses, ensuring that tracking does not get lost as long as the participant stays within the tracked area. The HTC Vive HMD has a 1080 × 1200 resolution per eye, with a 108° horizontal and a 97° vertical FOV. For the on-body tracking, we used the HTC Vive tracker (2018)[3], this tracker has a 270° FOV and

---

[1] https://docs.unity3d.com/ScriptReference/Time-deltaTime.html

[2] https://thearchitect4855.itch.io/unity-air-resistance

[3] https://dl.vive.com/Tracker/Guideline/HTC_Vive_Tracker_Developer_Guidelines_v1.3.pdf

was placed on the participant's dominant hand's wrist. As for the external sensor, we used Microsoft Kinect V2.0 because this sensor is off-the-shelf, widely known, and extensively used in research; this sensor has a 1920 × 1080 pixels, 30 FPS camera for color, a 512 × 424 pixels and 30 FPS sensor for depth, with a 70° × 60° FOV. This sensor was positioned 1.8m above the ground and at 3m in front of the participant's starting location. The position of the Kinect sensor was determined based on pilot tests to determine an area where the body tracking would be most accurate.

3.8 Participants

After receiving approval by our University's Institutional Review Board (IRB). We used G*Power for power analysis [12], and we selected a medium effect size, 15 conditions measurement with ANOVA within-subjects repeated measures mode, resulting in a minimum sample size of 24. We recruited 30 participants from our university, exchanging 50 min of their time for a $10 payment. Our final participant pool had 18 males, 11 females, and 1 non-binary participant. All participants were over 18 years old ($M = 21.14$, $SD = 3.49$). 29 participants were right-handed and 1 was left-handed. All participants spoke and comprehended English, walked and performed throwing gestures without assistance, and had normal or corrected to normal vision. No participant exhibited symptoms of visual, auditory, neurological, or physical disability. Participants self-reported their VR interaction frequency by selecting a number from 1 (never) to 5 (always) ($M = 1.94$, $SD = 0.97$). 7 participants practiced an activity or sport involving throwing gestures. Participants reported their height averaging $1.72m$ ($SD = 0.13$).

3.9 Procedure

Upon arrival at the study location, participants were greeted, given a detailed study protocol form, asked for consent, and evaluated for fitting the study requirements. Afterward, we collected their demographics through a survey. The study task was then verbally explained, and any questions were answered. Participants then wore the HMD and the HTC Vive tracker on their dominant hand's wrist. If any discomfort was expressed, we helped them adjust the apparatus until they were at ease. When unused, participants were asked to hold the VR controller in their non-dominant hand. The user study area dimensions were $4m \times 4m$, with the closest non-study physical object being 3m away.

The study started with recording the participant's arm span to calibrate the threshold values (see subsection 3.3) based on the original values in Table 1, proceeding to the training phase afterward. Participants practiced with each input configuration, throwable object, and throwing technique for 3 throws per condition, lasting approx. 10 minutes. The training helped familiarize participants with the input configurations, throwing metaphors, and sample target distances and reduced the learning effect during the main data collection portion. After training, we addressed any remaining questions and started the main VR experiment. We administered throwing conditions in a counterbalanced order (see subsection 3.6). Participants aimed at the current trial's target using the assigned configuration. The closest distance that the throwable object appeared at, away from the target's bullseye, was recorded as the accuracy metric for the active trial.

After each condition, participants answered in-VR survey questions on a Likert scale: **(1)** *Throwing direction*, **(2)** *Throwing speed*, **(3)** *Ease of adaptation*, **(4)** *Self-Reported performance*, **(5)** *Throwable correctly attached to hand*, **(6)** *Throwable correctly following of hand*, and **(7)** *Throwing realism* (see Table 4). The next set of trials only began after the survey completion. Each trial started from the same position (center of the tracked area), but participants could move or adjust themselves as needed to perform the throwing gesture. Once the full VR experiment was completed, participants completed a post-study survey about their throwing configuration preferences. They selected the overall most and least favorite configuration and explained their choices in a text entry field. While participants performed the VR task, the investigator was not in the play area, and no external noise or distractors were present.

4 RESULTS

Our results include objective accuracy and survey findings. We report the throwing accuracy in meters (m). We conducted a normality test using Shapiro-Wilk's test, indicating the data was normally distributed ($W = 0.956, p < .246$). We conducted an RM-ANOVA analysis to test the main and interaction effects between the factors involved in our experiment. We used Greenhouse-Geisser correction if Mauchley's sphericity test indicated a violation of the sphericity assumption. We used pairwise t-tests for our post-hoc analysis (see Table 2), and applied Bonferroni correction to protect against type 1 errors. To analyze the VR and post-study survey data, we used the Friedman non-parametric test along with Wilcoxon's Signed Rank test (see subsection 4.2). We provide bar plots for overall throwing accuracy results of throwing scenario (see Figure 4-(a)), configurations (see Figure 4-(b)), and scenario by configuration (see Figure 4-(c)).

Table 2: Input device configurations accuracy mean (meters) and standard deviations. CH: Controller Hold, CP: Controller Press, CT: Controller Threshold, VT: Vive Tracker, K: Kinect.

Device	Mean	Std. D
CH	1.179	.877
CP	1.154	.823
CT	1.219	.870
VT	1.053	.676
K	1.678	1.120

4.1 Main and Interaction Effects

We found a significant main effect of input configurations on throwing accuracy ($F_{4,116} = 27.971$, $p < .001$, $\eta_p^2 = .491$). Post-hoc analysis revealed a significant difference between *controller hold* vs *Vive tracker* ($t_{29} = 2.065$, $p < .048$), *controller hold* vs *Kinect* ($t_{29} = -6.444$, $p < .001$), *controller press* vs *Kinect* ($t_{29} = -6.310$, $p < .001$), *controller threshold* vs *Vive tracker* ($t_{29} = 3.345$, $p < .002$), *controller threshold* vs *Kinect* ($t_{29} = -7.517$, $p < .001$), and *Vive tracker* vs *Kinect* ($t_{29} = -9.187$, $p < .001$) (See full results in Table 3).

We found a significant main effect of the throwable object type on throwing accuracy ($F_{1.576,45.711} = 50.263$, $p < .001$, $\eta_p^2 = .634$). Post-hoc tests revealed a significant difference between them: *baseball*

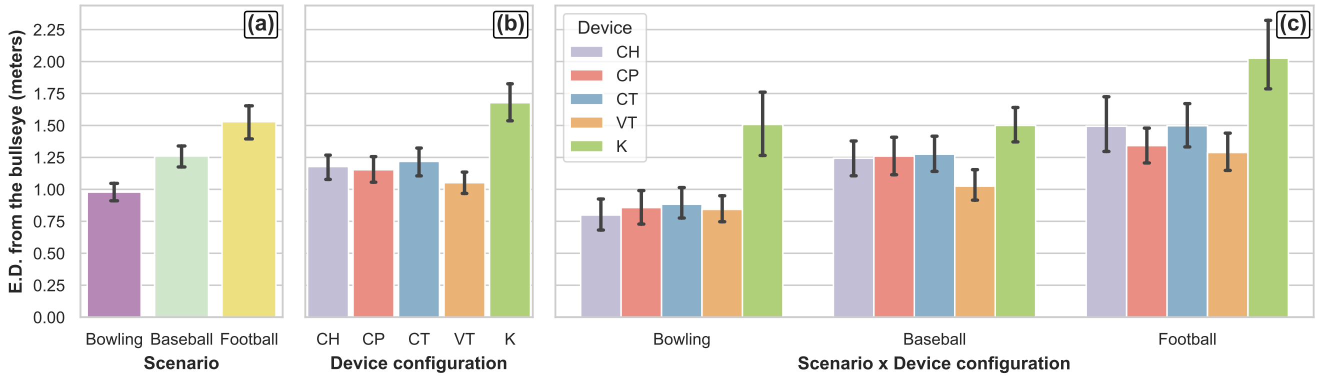

Figure 4: Throwing accuracy (average Euclidean distance from the bullseye) based on: (a) throwing scenario, (b) configuration, (c) scenario by configuration. CH: Controller Hold, CP: Controller Press, CT: Controller Threshold, VT: Vive Tracker, K: Kinect; lower values are better, ($CI = 95\%$).

vs *bowling* ($t_{29} = 6.608$, $p < .001$), *baseball* vs *football* ($t_{29} = -5.131$, $p < .001$), and *bowling* vs *football* ($t_{29} = -8.219$, $p < .001$). For interaction effects, no significant interaction between input configurations and throwable type on throwing performance was recorded ($F_{4.124,119.582} = 2.347$, $p < .057$, $\eta_p^2 = .075$).

Table 3: Pairwise T-Test for throwing configurations. CH: Controller Hold, CP: Controller Press, CT: Controller Threshold, VT: Vive Tracker, K: Kinect.

Pairs	t	df	Sig
CH - CP	.470	29	.642
CH - CT	-.701	29	.489
CH - VT	2.065	29	**.048**
CH - K	-6.444	29	**<.001**
CP - CT	-.981	29	.335
CP - VT	1.525	29	.138
CP - K	-6.310	29	**<.001**
CT - VT	3.345	29	**.002**
CT - K	-7.517	29	**<.001**
VT - K	-9.187	29	**<.001**

4.2 Qualitative Results

4.2.1 VR Survey. To analyze the Likert scale in-VR questionnaire, we averaged the scores per input configuration and ran Friedman non-parametric tests; statistical significance was shown for every question, so we ran additional Bonferroni-adjusted Wilcoxon Signed-Ranks tests. Here we present the results that we found interesting, while full results with significance and corresponding Z and P values for Wilcoxon Signed Rank tests are in Table 4 and Figure 6. The first interesting result was the participants' self-reported performance, where the controller press had the highest average score of 5.08, being significantly higher than the controller threshold at 4.50 ($Z = 56.5, p < .025$) and Kinect at 3.70 ($Z = 4, p < .001$). The second interesting finding came from the participant's perception of throwing realism, where the controller hold had the highest average score of 4.99, being only significantly higher than Kinect, which scored 3.63 ($Z = 12, p < .001$).

4.2.2 Post-study Survey. For the participants' most preferred throwing configuration, a Chi-squared test on the responses (χ_4^2 (N = 30) = 24, $p < .001$) indicated that the selected choices were not uniformly distributed. The percentages of choices were distributed in the following way: $CH = 53.33\%, CP = 23.33\%, CT = 10.00\%, VT = 10.00\%$, and $K = 3.33\%$ (number of votes are in Figure 5). For the participants' least preferred throwing configuration, a Chi-squared test on the responses (χ_4^2 (N = 30) = 24.67, $p < .001$) indicated that choices were not uniformly distributed. Percentages were as follows: $CH = 3.33\%$, $CP = 10.00\%$, $CT = 3.33\%$, $VT = 36.67\%$, and $K = 46.67\%$ (number of votes are in Figure 5).

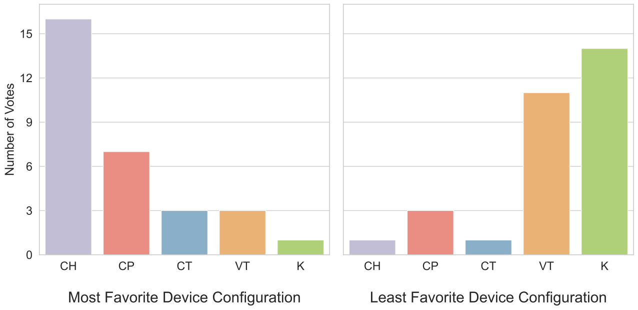

Figure 5: Number of votes for the most and least favorite throwing configurations. CH: Controller Hold, CP: Controller Press, CT: Controller Threshold, VT: Vive Tracker, K: Kinect. Controller hold was most and Kinect was least favored.

5 DISCUSSION

We aimed to find how different PoR mechanics using different input categories and throwable object types influence participants' performance and preference in throwing tasks. There were some discrepancies between perceived and actual performance, and participants' quotes aided us in understanding why these discrepancies exist (**RQ.5**). We also discuss some implementation challenges and how our decisions affected the user study outcomes.

5.1 Performance Results Implications

Hand-held and on-body devices (Vive tracker, controller hold, controller press, and controller threshold) performed similarly overall, among which the automatic PoR on-body device (Vive tracker) and the manual PoR hand-held device (controller press) performed better than others. Kinect performed significantly worse than all counterparts (all $ps < .001$) (see Figure 4-(b)). As shown in Figure 4-(a), performance in the bowling scenario was best, and performance in the football scenario was worst among the compared scenarios (**RQ.4**). This finding supports prior findings that underarm throwing is more accurate than overarm throwing [30].

One surprising result was that while throwing a bowling ball, the manual PoR hand-held device (controller hold) outperformed others, whereas, for football and baseball, the automatic PoR on-body device (Vive tracker) outperformed the others (see Figure 4-(c)). With these results, we suggest simultaneously using multiple PoR mechanics provided in Figure 1 along with different device categories. In our case, enabling controller hold for underarm throwing and Vive tracker for overarm throwing would be optimal. Performance with an automatic PoR external device (Kinect) was consistently and significantly worse than with others across all throwable object types. Specifically for throwing tasks, we suggest that researchers experiment with alternative external sensors rather than Kinect. Experimenting with adding devices such as Vicon motion capture and IMUs, along with other input device types, could be worthwhile.

5.2 Questionnaires Results Implications

In the in-VR survey, we asked seven questions, two of which were most interesting to discuss. The first one assessed self-reported performance after each trial, its responses indicating that by using the manual PoR hand-held device (controller press), participants felt more accurate, although it was only significantly different from automatic PoR hand-held device (controller threshold) and Kinect (**RQ.3**) (Figure 6). The second one assessed the overall realness after each trial, and the responses indicated that none of the throwing configurations except Kinect (underperformed) were significantly different and that controller hold had the highest average score for realness (**RQ.2**) (Figure 6). We believe that participants felt more accurate with controller press because they could minimize the timing inaccuracies and have control over the PoR, this also confirms prior findings about timing inaccuracies and complications [37].

In the post-VR survey, we asked the participants to choose their most and least favorite throwing configuration. According to participant preferences (Figure 5), controller hold was most favored by 16 participants, and 14 participants least favored Kinect (**RQ.1**). The Vive tracker was most favored by 3 participants and least favored by 11 participants, which is surprising, considering that it achieved the best objective average throwing accuracy among the compared throwing configurations (**RQ.5**). While we cannot directly explain the causes of the recorded discrepancies between objective and subjective performance, the reasons that participants provided with their choices offer insight into potential explanations.

Table 4: Friedman and Wilcoxon test results for overall data of the VR survey. CH: Controller Hold, CP: Controller Press, CT: Controller Threshold, VT: Vive Tracker, K: Kinect.

Question	Friedman		Wilcoxon		
	χ^2_4	P	Device	Z	P
1) In this scenario, was the throwing-direction as you expected it to be?	43.601	<.001	CH - K	16.5	<.001
			CP - K	10.5	<.001
			CT - K	45	.002
			K - VT	25	<.001
2) In this scenario, was the throwing-speed as you expected?	34.843	<.001	CH - K	23	.002
			CP - CT	30.5	.004
			CP - K	19	<.001
			K - VT	42.5	.012
3) How easy is it to adapt to this-throwing system in this scenario?	42.812	<.001	CH - CT	62	.039
			CH - K	21	<.001
			CP - K	30.5	.001
			CT - K	46	.01
			CT - VT	33.5	.008
			K - VT	8	<.001
4) How well did you perform using-this input configuration in this scenario?	48.499	<.001	CH - K	13.5	<.001
			CP - CT	56.5	.025
			CP - K	4	<.001
			CT - K	46.5	.004
			K - VT	0	<.001
5) In this scenario, Was the ball's-placement on your hand realistic?	43.675	<.001	CH - CT	44	.041
			CH - K	11.5	<.001
			CP - K	6.5	<.001
			CT - K	30.5	.001
			K - VT	32.5	.001
6) How well did the ball follow your-hand in this scenario (While attached)?	53.186	<.001	CH - CT	60.5	.033
			CH - K	3	<.001
			CP - CT	40	.048
			CP - K	8.5	<.001
			CT - K	2	<.001
			K - VT	5.5	<.001
7) Overall, when using this configuration, how accurate was the throwing compared to real-life?	40.424	<.001	CH - K	12	<.001
			CP - K	32.5	.002
			CT - K	28.5	.005
			K - VT	12.5	<.001

A common reason that participants provided for choosing controller hold as the most preferred throwing configuration is the feeling of being more in control of the exact PoR when throwing. One participant said: "*I was able to have a better knowledge of when I should release/let go*". Participants also stated that this configuration was more intuitive and closest to real-life throwing. Some participants provided reasons for the Vive tracker as their favorite, stating that it felt easier to use and that they were more accurate with it. However, 11 participants selected it as their least favorite configuration, providing reasons such as poor performance, difficulty with getting used to it, and the lack of realism due to not holding a real object with their physical hand. For Kinect, the primary reasons for being consistently selected as the least favorite configuration include the following: bad precision, the throwable object being jittery, control being difficult, and having the lowest accuracy.

In overarm throwing, the Vive tracker achieved the best accuracy across configurations, even though participants rated controller hold and press as more realistic and accurate. For controller threshold, the under-performance might be due to the small differences in how participants held the controller which, for thresholding,

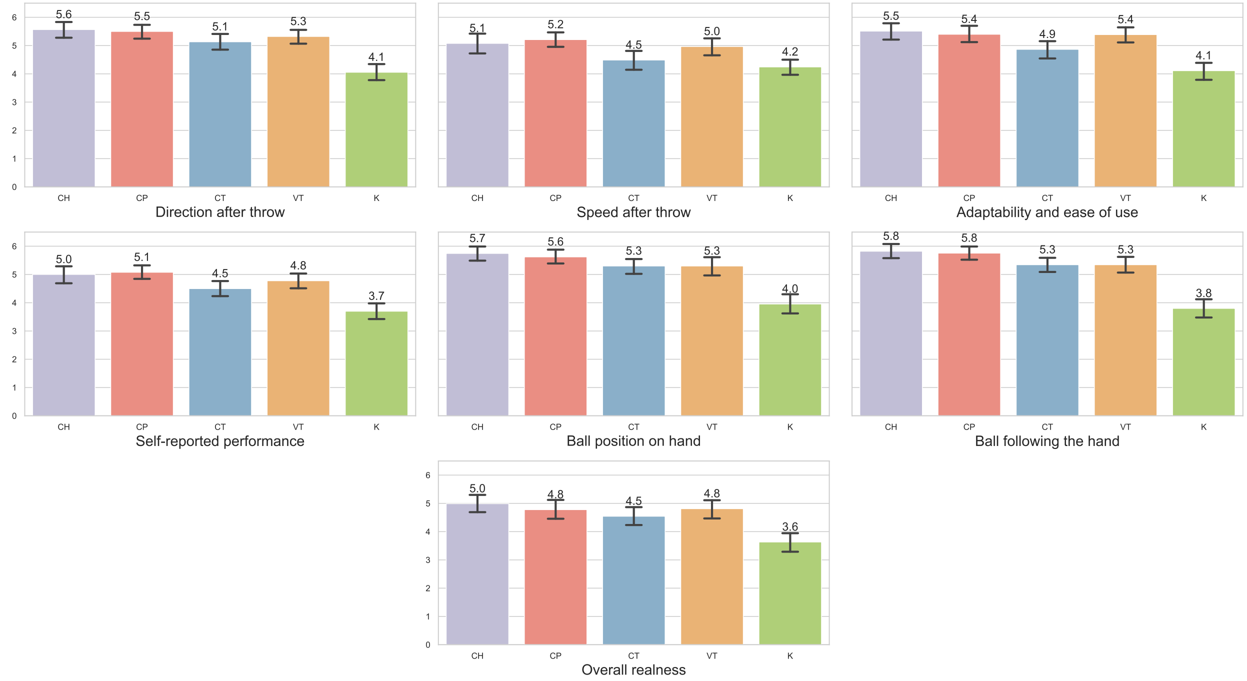

Figure 6: Average participant rating for the VR survey (higher score is better). CH: Controller Hold, CP: Controller Press, CT: Controller Threshold, VT: Vive Tracker, K: Kinect ($CI = 95\%$).

can affect the PoR and direction of throwing. Additionally, some participants (4) expressed being afraid of throwing the hand-held controller by mistake even though it was attached with a strap. We believe that another reason why Vive tracker was more accurate is that since it was attached to participants' hand and the PoR was automatic, they only had to focus on the direction and the speed of the throw, while for the manual PoR devices, they were not as accurate with timing the release.

5.3 Implementation Implications

During the implementation, throwing consistency across multiple configurations was a challenge. Vive tracker and Kinect do not have analog input for confirming the throwing PoR, so triggering the throwable release relied on an automatic approach based on the hand position. We applied the same threshold-based PoR mechanic to the controller threshold. There were multiple ways to implement thresholding. At an early stage in development, we implemented thresholding by receiving the spine and head joints' positions from the Kinect sensor, however due to the poor performance during pilot tests, we saw the need to replace this implementation. The improved implementation used the participant's hand position next to the head or hip as the origin, and the threshold defined the release trigger distance from that origin. The object release was triggered when the participant's hand crossed this specified distance threshold. We found this implementation to be effective when consistency across multiple device categories was required. We suggest defining a custom release distance threshold and using it in applications that revolve around a throwing mechanic.

6 LIMITATIONS AND FUTURE WORK

The scarcity of VR throwing research made it challenging to compare our results to findings and implementations in literature. More research and replication studies in this area are required to give additional context to our results. Overall, there are inherent differences between input configurations that are outside of our control (such as sampling rate and tracking accuracy) that prevent full standardization. We used the highest available refresh rate for each device, and to give each input configuration an equal chance of good performance, we adjusted the settings that worked best for each device (different number of frames to calculate the velocity, slightly different threshold values, double exponential smoothing filter for Kinect). Using the Kinect was challenging due to the noise in the data and before applying a smoothing filter there was a noticeable jitter hindering the throwing experience. On a few occasions, Kinect tracking was lost before the trial started, resulting in the throwable object not appearing on the participants' hand. In those instances, we asked the participants to exit and re-enter the tracked area so that Kinect tracking would be re-initialized and restarted the trial. As options for external tracking input configuration, during prototyping we considered UltraLeap, however, it has a limited FOV of (170°), making it not suitable for throwing, where the hand frequently exits the tracking at the PoR. Cameras from the RealSense series present another alternative, however, unlike Kinect, they are more fitted for finger tracking and have the same FOV issue as UltraLeap. Since RealSense cameras are generally used for other tasks, such as environment scanning and drone navigation, Kinect is known to outperform them because it has a specialized solution for body tracking. Monocular camera solutions operate in pixel space, presenting issues with depth and scaling, both of which are important for accurate throwing in VR. A possible solution to all these problems would involve a multi-camera setup. This would likely involve camera calibration and training a model that would merge joint positions from multiple angles, providing accurate depth values. However, this could be worthwhile because if accurate enough, participants may prefer an input configuration that eliminates the need of hand-held or on-body devices.

Since VEs with different levels of visual cues influence spatial perception [9, 20], we plan to try different types of VEs to evaluate their influence on throwing accuracy and speed. Incorporating dynamic targets is also a logical next step to assess throwing accuracy in such settings [35]. We aim to evaluate more methods for computing throwing velocity, such as using only the last two frames, rigid body velocity, or only using the maximum reached speed across frames [37], in a study that varies throwing configurations. This will contribute to the set of VR-centered throwing implementation guidelines. Conducting similar experiments in augmented reality (AR) or mixed reality (MR) is interesting, considering that research regarding throwing metaphors is under-explored across the extended reality (XR) spectrum. We note that throwing repeatedly for many consecutive trials requires physical effort, so participants can become fatigued in later trials. In our study, we used the in-VR survey after each condition (6 trials) as a rest interval, and we suggest future VR throwing studies also incorporate rest time intervals or low-effort activities to limit participant fatigue.

7 CONCLUSION

We investigated throwing different virtual object types alongside using different PoR mechanics in VR accompanied by qualitative data from 30 participants. Our analysis reveals multiple interesting findings. Four of five input configurations led to similar performance, with Vive tracker resulting in the best overall accuracy. The bowling surpassed the other throwable objects, meaning that participants were more accurate with underarm throwing. Findings from the post-study survey indicate that participants preferred controller hold, which is the commonly used manual PoR VR controller-based interaction method. Participants felt more in control of the PoR and overall throwing gesture, even though this throwing configuration was not the most accurate. Throwing using Kinect sensor led to overall poor results across the entire study, and it was the least preferred throwing configuration by the participants. We showed that when PoR thresholding is used properly, the achieved results are similar, and in some cases even better than using traditional manual PoR.

ACKNOWLEDGMENTS

This work is supported in part by NSF Award IIS-1917728, Northrop Grumman, Unknot.id, and the Florida High Tech Corridor Council Industry Matching Research Program. We also thank the anonymous reviewers for their insightful feedback and the ISUE lab members for their support.

REFERENCES

[1] [n. d.]. Throwing objects is frustratingly broken. :: SUPERHOT VR General Discussions. https://steamcommunity.com/app/617830/discussions/0/3086646248549197886/

[2] Michael Bonfert, Stella Lemke, Robert Porzel, and Rainer Malaka. 2022. Kicking in Virtual Reality: The Influence of Foot Visibility on the Shooting Experience and Accuracy. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 711–718. https://doi.org/10.1109/VR51125.2022.00092

[3] Michael Bonfert, Stella Lemke, Robert Porzel, and Rainer Malaka. 2022. Kicking in Virtual Reality: The Influence of Foot Visibility on the Shooting Experience and Accuracy. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, Christchurch, New Zealand, 711–718. https://doi.org/10.1109/VR51125.2022.00092

[4] Malte Borgwardt, Jonas Boueke, María Fernanda Sanabria, Michael Bonfert, and Robert Porzel. 2023. VRisbee: How Hand Visibility Impacts Throwing Accuracy and Experience in Virtual Reality. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–7.

[5] Lai Lila Bozgeyikli and Evren Bozgeyikli. 2022. Tangiball: foot-enabled embodied tangible interaction with a ball in virtual reality. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 812–820.

[6] copper_tunic. 2024. Throwing in VR. www.reddit.com/r/virtualreality/comments/19e1m3q/throwing_in_vr/

[7] Alexandra Covaci, Anne-Hélène Olivier, and Franck Multon. 2015. Visual Perspective and Feedback Guidance for VR Free-Throw Training. *IEEE Computer Graphics and Applications* 35, 5 (2015), 55–65. https://doi.org/10.1109/MCG.2015.95

[8] Alexandra Covaci, Cristian-Cezar Postelnicu, Alina Ninett Panfir, and Doru Talaba. 2012. A Virtual Reality Simulator for Basketball Free-Throw Skills Development. In *Technological Innovation for Value Creation*, Luis M. Camarinha-Matos, Ehsan Shahamatnia, and Gonçalo Nunes (Eds.). Vol. 372. Springer Berlin Heidelberg, Berlin, Heidelberg, 105–112. https://doi.org/10.1007/978-3-642-28255-3_12 Series Title: IFIP Advances in Information and Communication Technology.

[9] Sarah H Creem-Regehr, Jeanine K Stefanucci, and Bobby Bodenheimer. 2023. Perceiving distance in virtual reality: theoretical insights from contemporary technologies. *Philosophical Transactions of the Royal Society B* 378, 1869 (2023), 20210456.

[10] Stefanie A Drew, Madeline F Awad, Jazlyn A Armendariz, Bar Gabay, Isaiah J Lachica, and Jacob W Hinkel-Lipsker. 2020. The trade-off of virtual reality training for dart throwing: a facilitation of perceptual-motor learning with a detriment to performance. *Frontiers in Sports and Active Living* 2 (2020), 59.

[11] Fletcher Dunn and Ian Parberry. 2011. *3D math primer for graphics and game development*. CRC Press.

[12] Franz Faul, Edgar Erdfelder, Axel Buchner, and Albert-Georg Lang. 2009. Statistical power analyses using G* Power 3.1: Tests for correlation and regression analyses. *Behavior research methods* 41, 4 (2009), 1149–1160.

[13] kaitoofrose. 2021. Anyone else having trouble throwing in VR? www.reddit.com/r/virtualreality/comments/rd2r3i/anyone_else_having_trouble_throwing_in_vr/

[14] Jindrich Kodl, Albert Mukovskiy, Tjeerd Dijkstra, Doris Brötz, Nicolas Ludolph, Nick Taubert, and Martin Giese. 2017. Ball Throwing Games in Virtual Reality for Motor Rehabilitation.

[15] Julian Kreimeier, Sebastian Hammer, Daniel Friedmann, Pascal Karg, Clemens Bühner, Lukas Bankel, and Timo Götzelmann. 2019. Evaluation of different types of haptic feedback influencing the task-based presence and performance in virtual reality. In *Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments*. ACM, Rhodes Greece, 289–298. https://doi.org/10.1145/3316782.3321536

[16] kyboon. 2021. How do you think throwing in VR is? Is it accurate and feels nature? www.reddit.com/r/virtualreality/comments/pg0i3i/how_do_you_think_throwing_in_vr_is_is_it_accurate

[17] Joseph J. LaViola. 2003. Double Exponential Smoothing: An Alternative to Kalman Filter-Based Predictive Tracking. In *Proceedings of the Workshop on Virtual Environments 2003* (Zurich, Switzerland) *(EGVE '03)*. Association for Computing Machinery, New York, NY, USA, 199–206. https://doi.org/10.1145/769953.769976

[18] Joseph J LaViola Jr, Ernst Kruijff, Ryan P McMahan, Doug Bowman, and Ivan P Poupyrev. 2017. *3D user interfaces: theory and practice.* Addison-Wesley Professional.

[19] José-Antonio Lozano-Quilis, Sergio Albiol-Pérez, Hermenegildo Gil-Gómez, Guillermo Palacios-Navarro, Habib Fardoun, José-Antonio Gil-Gómez, and Abdulfattah Mashat. 2013. Virtual Reality System for Multiple Sclerosis Rehabilitation using KINECT. In *Proceedings of the ICTs for improving Patients Rehabilitation Research Techniques*. IEEE, Venice, Italy. https://doi.org/10.4108/icst.pervasivehealth.2013.252208

[20] Sina Masnadi, Yahya Hmaiti, Eugene Taranta, and Joseph J LaViola Jr. 2023. Effects of Clutter on Egocentric Distance Perception in Virtual Reality. *arXiv preprint arXiv:2304.08604* (2023).

[21] MowTin. 2023. Throwing grenades in VR: Has anyone solved this? www.reddit.com/r/virtualreality/comments/1472s35/throwing_grenades_in_vr_has_anyone_solved_this/

[22] Ok-Guess-9059. 2024. How to throw very far in Superhot VR? www.reddit.com/r/virtualreality/comments/1am2cqr/how_to_throw_very_far_in_superhot_vr/

[23] Yogesh Singh, Sunil K Agrawal, and Vineet Vashista. 2022. Throwing strategy in a dual-motor-task of aiming at the bullseye while walking in virtual reality. *IEEE Robotics and Automation Letters* 7, 4 (2022), 9091–9098.

[24] Yogesh Singh, Antonio Prado, Dario Martelli, Fitsum E Petros, Xupeng Ai, Sudipto Mukherjee, Anil K Lalwani, Vineet Vashista, and Sunil K Agrawal. 2020. Dual-motor-task of catching and throwing a ball during overground walking in virtual reality. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 28, 7 (2020), 1661–1667.

[25] Teh1Person0. 2022. VR Throwing is a key part of our game, but it frustrates a lot of our players, so we are trying some alternatives. Any suggestions? www.reddit.com/r/VRGaming/comments/wskp69/vr_throwing_is_a_key_part_of_our_game_but_it/

[26] Judith Tirp, Christina Steingröver, Nick Wattie, Joseph Baker, and Jörg Schorer. 2015. Virtual realities as optimal learning environments in sport-A transfer study of virtual and real dart throwing. *Psychological Test and Assessment Modeling* 57, 1 (2015), 57.

[27] Yuki Ueyama and Masanori Harada. 2022. Effects of first- and third-person perspectives created using a head-mounted display on dart-throwing accuracy. *Virtual Reality* 26, 2 (June 2022), 687–695. https://doi.org/10.1007/s10055-021-00562-x

[28] Sam Van Damme, Jordy Tack, Glenn Van Wallendael, Filip De Turck, and Maria Torres Vega. 2023. Are we ready for Haptic Interactivity in VR? An Experimental Comparison of Different Interaction Methods in Virtual Reality Training. In *2023 15th International Conference on Quality of Multimedia Experience (QoMEX)*. 294–299. https://doi.org/10.1109/QoMEX58391.2023.10178453

[29] Marvin Winkler and Stefan M Grünvogel. 2023. Influence of Simulated Aerodynamic Forces on Weight Perception and Realism in Virtual Reality. In *2023 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 811–812.

[30] Goksu Yamac, Jackey JK Chang, and Carol O'Sullivan. 2023. Let it go! Point of release prediction for virtual throwing. *Computers & Graphics* 110 (2023), 11–18.

[31] Goksu Yamac, Niloy J. Mitra, and Carol O'Sullivan. 2021. Detecting the Point of Release of Virtual Projectiles in AR/VR. In *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. 563–564. https://doi.org/10.1109/VRW52623.2021.00164

[32] Goksu Yamac and Carol O'Sullivan. 2022. Eye on the Ball: The Effect of Visual Cue on Virtual Throwing. In *SIGGRAPH Asia 2022 Posters* (Daegu, Republic of Korea) *(SA '22)*. Association for Computing Machinery, New York, NY, USA, Article 50, 2 pages. https://doi.org/10.1145/3550082.3564181

[33] Goksu Yamac and Carol O'Sullivan. 2022. FauxThrow: Exploring the effects of incorrect point of release in throwing motions. In *ACM Symposium on Applied Perception 2022*. 1–5.

[34] Zhengwei Yao, Yimin Chen, Ming Chen, and Yongshun Xu. 2008. Throwing Recognition Based on Magnetic Tracking and Trajectory Computation in an Augmented Reality System. In *2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*. IEEE, Wuhan, China, 543–547. https://doi.org/10.1109/PACIIA.2008.208

[35] Difeng Yu, Brandon Victor Syiem, Andrew Irlitti, Tilman Dingler, Eduardo Velloso, and Jorge Goncalves. 2023. Modeling Temporal Target Selection: A Perspective from Its Spatial Correspondence. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, Hamburg Germany, 1–14. https://doi.org/10.1145/3544548.3581011

[36] Zhaoran Zhang and Dagmar Sternad. 2021. Back to reality: differences in learning strategy in a simplified virtual and a real throwing task. *Journal of Neurophysiology* 125, 1 (Jan. 2021), 43–62. https://doi.org/10.1152/jn.00197.2020

[37] Tim Zindulka, Myroslav Bachynskyi, and Jörg Müller. 2020. Performance and experience of throwing in virtual reality. In *Proceedings of the 2020 CHI conference on human factors in computing systems*. 1–8.