Perception Hacking for 2D Cursorjacking in Virtual Reality

Zihao Su
zs3pv@virginia.edu
University of Virginia

Faysal Hossain Shezan
fs5ve@virginia.edu
University of Virginia

Yuan Tian
yt2e@virginia.edu
University of Virginia

David Evans
evans@virginia.edu
University of Virginia

Seongkook Heo
seongkook@virginia.edu
University of Virginia

ABSTRACT

Virtual reality offers an immersive virtual environment in which users interact with virtual objects relying on visual and auditory feedback rendered by the system. This creates new opportunities to manipulate the user's perception by exploiting the discrepancies between the virtual and physical hands, which can produce various haptic effects and increase immersiveness. This paper calls attention to the potential uses of perception hacking to mislead users to inadvertently select malicious 2D targets. We demonstrate the hazards of adversarial perception hacking by implementing a cursorjacking attack technique. The attack gradually deviates the displayed cursor from the actual controller movement, causing users to make unintended adjustments to the controller movement and select unintended targets. Our preliminary experiment found that participants could easily be fooled into generating clicks on different objects from the one they intended without being aware that they were being manipulated.

CCS CONCEPTS

• **Human-centered computing → Virtual reality**.

KEYWORDS

Cursorjacking, clickjacking, virtual reality, perception hacking

ACM Reference Format:
Zihao Su, Faysal Hossain Shezan, Yuan Tian, David Evans, and Seongkook Heo. 2022. Perception Hacking for 2D Cursorjacking in Virtual Reality. In *CHI Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 5 pages.

1 INTRODUCTION

Virtual Reality (VR) is an immersive computing platform that opens up many possibilities to enhance user experiences by manipulating user perception, such as creating illusions of physical forces [22], varying sizes and shapes [1], and causing redirected movements [20]. This immersiveness also brings unique security challenges in ensuring the integrity of their interactions. Due to the separation of the virtual and physical world, identifying manipulations in the interactions is more challenging in VR than in other scenarios where users can see the physical movements of the input device. This becomes even more difficult with the input adjustment techniques implemented in VR [4, 8] to support more stable and accurate mid-air input [5]. Scenarios that involve degree-of-freedom (DOF) reduction, such as browsing the web in VR, will have another component that translates the 6-DOF controller input into 2D movement. This may expose users to perception-based attacks where users are tricked into taking some unintended action in the sensitive 2D UI, which is known as *clickjacking*.

Clickjacking is a well-known UI deception attack, most often exploited on web browsers [11]. It entails an attacker tricking a user to click on a button that is different from what the user intends. It is effective in scenarios where a user click can authorize certain sensitive actions that the attacker cannot achieve by themselves, such as when an attacker controls a script embedded in an iframe on a webpage. One technique used for clickjacking is known as *cursorjacking* that shows a fake cursor to the user to provide misleading cues of the cursor position [10].

While web clickjacking has been well studied and defenses are now widely deployed [10, 11, 15, 21, 23], it has not been well explored in the VR system. This paper highlights how the immersiveness of VR introduces potential risks to clickjacking by demonstrating and analyzing a cursorjacking attack technique that exploits the visual dominance of human perception in judging hand movements [13]. Humans weigh visual perception more than other senses when there is a disparity between them. Prior studies have shown that if the disparity is small, participants do not notice the disparity while moving their hands following the manipulated path guided by visual movement [3, 12]. Similarly to how prior work used the angular offset to redirect hand movement in VR [3], the demonstrated attack technique adds intentional offsets to a cursor visualization during cursor movement, tricking users into making adjustments to the cursor to compensate for the offset.

To investigate the opportunity for such an attack technique to induce targeted movements without user awareness, we conducted a preliminary experiment with 23 participants. In the experiment, participants completed 2D pointing tasks using a VR controller, consecutively selecting two targets displayed on a canvas. The displayed cursor's movement was manipulated by the angle between 0 and 25 degrees in both clockwise and counterclockwise directions. The results showed that the attack technique could successfully lead participants to select a malicious target without noticing any differences between conditions without manipulation and 10 degrees or less manipulation. Even with 25 degree manipulation, participants failed to reliably detect the manipulation.

CHI EA '22, Proceedings of the 1st Workshop on Novel Challenges of Safety, Security and Privacy in Extended Reality, April 29-May 5, 2022, New Orleans, LA, USA

In this paper, we argue that the effectiveness of perception hacking in VR enables stealthy input-related attacks like clickjacking in VR, as attackers have powerful opportunities to deceive users' perception and manipulate user behavior without users being aware that they are being manipulated. We speculate on scenarios in which clickjacking attacks could be used to accomplish adversarial goals and incur real harm.

2 RELATED WORK

Prior work [9, 26, 27] explored different aspects of security, privacy, and safety risk assessment for VR systems. Casey et al. introduced an attack in VR [7], in which the manipulated surroundings (e.g., safety boundary, world orientation) can physically harm users. Rafique et al. explored an attack technique by blocking and manipulating VR tracking system [19]. Andrade et al. [2] and Ling et al. explored attacks that steal sensitive information through side channeling that use VR hardware data. However, none of these focused on the security of the UI interactions in VR. Lee et al. discovered the lack of isolation protection against third-party scripts in WebVR and proposed multiple Ad fraud attacks in WebVR, including gaze and controller cursorjacking attacks, which assumed an attacker that controls embedded third-party scripts to be capable of directly manipulating the cursor and the hosting website's environment [14]. In contrast, our work proposes a cursorjacking attack that assumes that the attacker has no permission to directly perform sensitive actions in embedded third-party overlays and utilizes perception hacking techniques to steal user click.

Our approach is inspired by studies on manipulating user perceptions in VR for immersive experiences [3, 20, 22, 24]. This is often done by manipulating the Control-Display Ratio (CD Ratio) of object movements or scene transitions to make visual cues of virtual objects or scenes discrepant from proprioceptive cues. Since visual cues dominate human perception when the senses provide conflicting stimuli [6, 25], this discrepancy can be imperceptible to users, and illusions of realistic experiences can be created. Previous work [3, 20, 22, 24] manipulated the CD ratio to create realistic illusions for different scenarios in VR, such as illusions of walking in a large room [20, 24] or illusions of object weight [22]. We use a similar technique of gradually shifting the visual cursor in our cursorjacking attack.

3 CURSORJACKING ATTACK

3.1 Attack Assumption

Our attack technique method assumes a two-button setup. Firstly, we assume that there is a desired *target button*, a button that the attacker wants the user to click, and a button that the users want to click (i.e., *bait button*). Lastly, we also assume that the application knows the positions of the two buttons. Note that in contrast to the classic cursorjacking attacks, the positions here are 3D coordinates in the app's virtual environment.

Under this setup, we assume that an attacker is motivated to lure the user to click on *bait button* and to hijack this click to *target button*, possibly because *target button* can authorize a sensitive action which the attack cannot initiate. In our study, *target button* and *bait button* were placed on the same 2D plane in our prototype for simplicity. However, in an actual attack scenario, they do not need to be on the same virtual object/plane in VR.

3.2 Attack Technique

Our attack technique was designed with two considerations: 1) how to manipulate the user to click on an arbitrary location and 2) how to make the manipulation undetectable.

First, to manipulate the user to click on an arbitrary location, our technique induces users to make an attacker-intended adjustment to the cursor using manipulated visual cues (see Figure 1). This is based on the findings from prior studies showing that users have an adjustment phase during target acquisition tasks [17, 18]. That is, when a user makes a target selection movement, they often cannot move perfectly towards the intended button by proprioception, resulting in adjusting the movement based on visual cues. Studies have also shown that when there is a discrepancy between what a user sees (visual cue) and what the user feels from their muscles and joints (proprioceptive cue), visual cues dominate [6, 25]. As a result, during cursor manipulation, false visual cues for the cursor can lead the user to make corresponding false adjustments, even if the visual cues are somewhat discrepant from the user's proprioceptive cues. This attack adds intentional offsets to the displayed cursor over time, so the user is induced to make adjustment to the cursor position to counter the offsets and control the displayed cursor to move towards their intended button.

Second, if the offset created by the attacker is intrusive and obvious, the user may not ascribe the offset to be of their action and may suspect the integrity of the cursor. To make the manipulation undetectable, we designed an adaptive attack that applies small manipulations to the displayed cursor in real time only when the user moves the cursor, so that the manipulations are blended into the user's cursor movement. Moreover, we controlled the angle of manipulation and evaluated to what extent users can detect the manipulation for varying degrees of manipulation, thus understanding the range for which the manipulation is stealthy and undetectable.

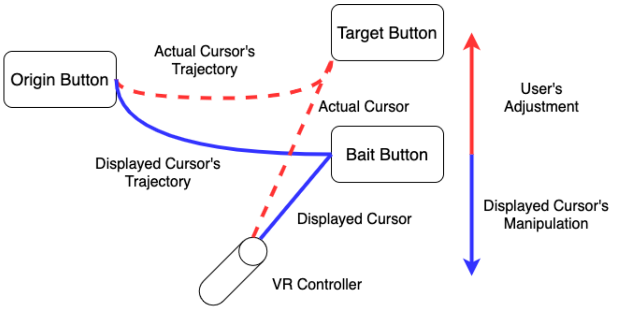


Figure 1: Cursor manipulation mechanism. During the cursor movement towards bait button, gradually moving the displayed cursor towards the opposite direction to the attacker's intended target can lead the user to make corrective movement towards the target without noticing.

3.3 Attack Implementation

An overview of our implementation of the attack technique is as follows:

(1) **Attack initialization.** Once attack is initiated (in our experiment, origin button clicked), the attack will aim to create an offset (referred to as "Target Vector" below) to the displayed cursor that is equal to the vector from the target button to the bait button. This aimed offset is used to induce the user to make attacker-intended adjustment to the cursor.

(2) **Attack process.** At every time frame ($\frac{1}{60}$ in our setup), the program measures the cursor movement during that frame. Then, a small offset vector in the direction of Target Vector is added to the displayed cursor, with the magnitude constrained to a small fraction (which we refer to as "Threshold Ratio") of the movement distance of the actual cursor. This constraint is applied to limit the discrepancy between the user's visual cues of the cursor and the user's proprioceptive cues of the hand movement and to make the manipulation undetectable.

(3) **Termination.** We terminate an attack when the *target button* has been clicked (i.e., the attack has been successfully commenced). If the program decides to terminate the attack, it reversely manipulates the displayed cursor back to the actual cursor's position as the user moves the cursor to prevent it from being noticed.

4 EXPERIMENT

The presented attack technique utilizes difficulties in perceiving discrepancies between the movement of the controller and the visual cursor. Therefore, it is crucial to understand how a different level of discrepancy affects users' perception of cursor movement and from which point they begin to feel the cursor movement to be abnormal. We conducted an experiment to identify the relationship between the maximum manipulation angle and the detection ratio.

4.1 Experiment Design

We recruited 23 participants through the local university online forum and student groups on social media. The average age was 25.9 years old (20–35, SD=4.87). The experiment took around 20 minutes for each participant. The experiment protocol has been reviewed and approved by the UVA IRB (#2878).

Participants sat on a chair while wearing an Oculus Rift S VR headset. They completed tasks using an Oculus Touch controller while resting their arms on the armrest to prevent fatigue. The experiment application was implemented using Unity and ran on a PC with an Intel Core i7 processor and NVidia RTX 2080Ti GPU.

Our experiment used a repetitive target selection task that varied the target orientation each trial, while keeping the distance consistent to prevent confounding. In the experiment, a 2D screen with three GUI buttons, which were origin, target, and bait buttons, was shown in the 3D space. The origin button was placed in the middle of the screen, and the target and bait buttons were placed at a random location with a fixed distance from the origin button. The screen was 5 meters high and 8 meters wide and was placed at 2 meters away from the participant in VR. The size of each button was $0.2m \times 0.2m$. Participants were asked to complete target selection tasks with a 2D cursor on the screen controlled by the VR controller. The actual mouse cursor location was determined by projecting the ray from the controller (i.e., ray casting), however, the participants were only shown a manipulated cursor (i.e., displayed cursor) that could be shifted based on the study condition. While the actual cursor was hidden, all UI interactions were processed based on the actual cursor's click location.

The independent variable used in the experiment was the angle of manipulation, which determines the angular separation between the bait button and the target button, therefore determining the needed offset between the displayed cursor and the actual cursor (v_t). We used 11 different angles of manipulation: 0 (no manipulation), $\pm5^\circ$, $\pm10^\circ$, $\pm15^\circ$, $\pm20^\circ$, $\pm25^\circ$. The positive degree angle means that the target button was placed in the counterclockwise direction relative to the bait button, and the negative degree angle means the opposite. Participants completed 10 trials for each angle of manipulation (110 trials in total), and the order of the trials was randomized to reduce the ordering effect.

In the experiment, participants were asked to click the origin button and then the bait button. Before actual trials, participants were asked to complete 10 trials without cursor manipulation to familiarize themselves with normal cursor movement. They were informed that the movement they experienced was normal and asked to remember it. During the study, the attack algorithm manipulated the movement of the displayed cursor between the two clicks. After each trial, participants were asked to answer if they felt the cursor movement between the buttons was weird or normal. After completing all tasks, each participant received $15.

4.2 Results

Figure 2 shows the proportion of responses where participants felt the movement was not normal, by cursor manipulation angle. The curves were fitted using the following psychometric function [1]:

$$f(x) = \frac{1}{1 + e^{ax+b}} \tag{1}$$

where a and b are real numbers. The proportion of non-normal movement response of 1 means all participants recognize all trials to be not normal, and the proportion of 0 means all perceive the movement to be normal. We observe that participants even perceived the manipulated movement with a 5-degree angle to be more 'normal' than without any manipulation, while the difference can be negligible (difference in mean <0.014%). When the angle of manipulation was 10 degrees or smaller, the participants perceived them as non-normal less than 15% of the time, regardless of the manipulation direction. As shown in Figure 3, the perception of different angles of manipulation was different across participants, especially when the angle of manipulation was larger. While the difference became smaller as the angle of manipulation decreased, however, the difference was still large when there was no manipulation (IQR=0.2).

5 DISCUSSION

Our experiment suggests that VR environments can be designed to manipulate user actions in harmful and unnoticeable ways. For the proportion of detecting non-normal movements, researchers use 75% as the detection threshold of a stimulus [16]. Yet, as seen in

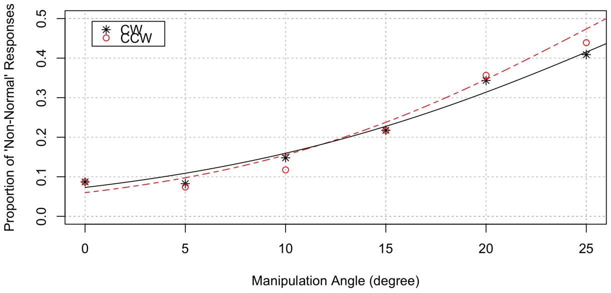


Figure 2: Proportion of 'Non-Normal' Responses by Angle of Cursor Manipulation. The lines show the psychometric function fitted curves.

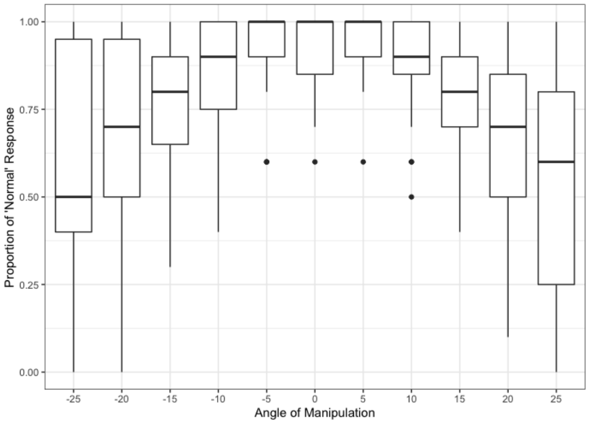
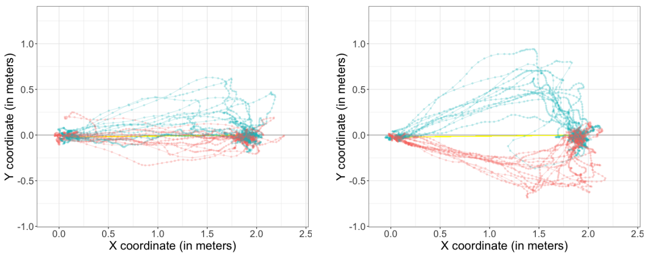


Figure 3: Proportion of "Normal" Response for Different Angles of Manipulation.

figure 3, none of the manipulated conditions with different angles was detected more than 50% of the time at the median, indicating that the participants were unable to reliably tell if there was cursor manipulation. When the angle of manipulation was 10 degree or less, the result implies that the participants were confident that the cursor movement felt normal, even though they were repeatedly asked about it. It is likely that if participants face such manipulation in a malicious app, they would not report suspicious behaviors.

In Figure 4, users' cursor movements present more curved trajectories as the angle of manipulation increases, both for clockwise (turquoise lines) and counterclockwise (orange lines) manipulations, which means that they should feel a larger discrepancy between visual and proprioceptive cues. However, the participants still perceived the cursor movements as normal in all trials corresponding to the data in the figure. This result is likely because the users are completely blocked from real-world movement, unable to see the movement of their hands and controllers, leading to not noticing strong cursor manipulations.

The results show that users are put in a vulnerable position and that they are likely unable to perceive malicious behaviors if malicious visual manipulations are involved in a VR application. Therefore, it is important to develop a mechanism to detect such a manipulation and prevent users from being harmed by the manipulation. We envision several potential options to defend against our proposed attack. Similarly to classic defenses of clickjacking attacks on desktop browsers, the VR system can also ensure the *context integrity* of sensitive functionalities to prevent our proposed attack [10], making sure that the user sees everything they should see before clicking on sensitive buttons. Moreover, as our observation above shows a clear difference in cursor trajectories when manipulations are in place, detection of abnormal cursor movement patterns can also be implemented via training models on the cursor movements. However, the tactics above would require changes to the current VR system, because they need the information of UIs (e.g. HTML structure for the classic defenses) and/or pointers, but such information of individual VR apps is not readily available to the VR system. Lastly, enforcing a standardized cursor across all VR apps will enable a more secure control of cursors, but this action would compromise the usability of many VR apps, because they usually implement their own cursors to create different ways for the users to interact with the APP.



(a) 5° of manipulation

(b) 25° of manipulation

Figure 4: Cursor trajectory with 5° and 25° manipulation.

6 CONCLUSION

In this paper, we demonstrated potentials for adversarial perception hacking. We believe that the potential threat would be greater in the real-life attack, because in our experiment, participants were already aware of the existence of abnomal cursor movement and were asked to actively find them. Even in such a setting, participants could not reliably detect manipulated cursors. Our study did not fully explore all factors that can influence the user's perception of the cursor movement, such as the speed of the cursor or the distance to the target.

Although this work demonstrates the effectiveness of the potential attack, we still have a question about how this attack could be built into an actual application, which is essential to evaluate the potential impact of this attack and to develop defense mechanisms. We hope to discuss this at the workshop.

REFERENCES

[1] Parastoo Abtahi and Sean Follmer. 2018. Visuo-haptic illusions for improving the perceived performance of shape displays. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.

[2] Tiago Martins Andrade, Max Smith-Creasey, and Jonathan Francis Roscoe. 2020. Discerning User Activity in Extended Reality Through Side-Channel Accelerometer Observations. In *2020 IEEE International Conference on Intelligence and Security Informatics (ISI)*. IEEE, 1–3.

[3] Mahdi Azmandian, Mark Hancock, Hrvoje Benko, Eyal Ofek, and Andrew D Wilson. 2016. Haptic retargeting: Dynamic repurposing of passive haptics for enhanced virtual reality experiences. In *Proceedings of the 2016 chi conference on human factors in computing systems*. 1968–1979.

[4] Marc Baloup, Thomas Pietrzak, and Géry Casiez. 2019. Raycursor: A 3d pointing facilitation technique based on raycasting. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12.

[5] Anil Ufuk Batmaz, Mohammad Rajabi Seraji, Johanna Kneifel, and Wolfgang Stuerzlinger. 2021. No Jitter Please: Effects of Rotational and Positional Jitter on 3D Mid-Air Interaction. In *Proceedings of the Future Technologies Conference (FTC) 2020, Volume 2*, Kohei Arai, Supriya Kapoor, and Rahul Bhatia (Eds.). Springer International Publishing, Cham, 792–808.

[6] E. Burns, S. Razzaque, A.T. Panter, M.C. Whitton, M.R. McCallus, and F.P. Brooks. 2005. The hand is slower than the eye: a quantitative exploration of visual dominance over proprioception. In *IEEE Proceedings. VR 2005. Virtual Reality, 2005*. 3–10. https://doi.org/10.1109/VR.2005.1492747

[7] Peter Casey, Ibrahim Baggili, and Ananya Yarramreddy. 2019. Immersive virtual reality attacks and the human joystick. *IEEE Transactions on Dependable and Secure Computing* (2019).

[8] Géry Casiez, Nicolas Roussel, and Daniel Vogel. 2012. 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*. 2527–2530.

[9] Aniket Gulhane, Akhil Vyas, Reshmi Mitra, Roland Oruche, Gabriela Hoefer, Samaikya Valluripally, Prasad Calyam, and Khaza Anuarul Hoque. 2019. Security, Privacy and Safety Risk Assessment for Virtual Reality Learning Environment Applications. In *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 1–9.

[10] Lin-Shung Huang, Alex Moshchuk, Helen J Wang, Stuart Schecter, and Collin Jackson. 2012. Clickjacking: Attacks and defenses. In *Presented as part of the 21st {USENIX} Security Symposium ({USENIX} Security 12)*. 413–428.

[11] Ilya Kantor. 2019. The clickjacking attack. https://javascript.info/clickjacking.

[12] Luv Kohli, Mary C Whitton, and Frederick P Brooks. 2012. Redirected touching: The effect of warping space on task performance. In *2012 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 105–112.

[13] Anatole Lécuyer, Sabine Coquillart, Abderrahmane Kheddar, Paul Richard, and Philippe Coiffet. 2000. Pseudo-haptic feedback: can isometric input devices simulate force feedback?. In *Proceedings IEEE Virtual Reality 2000 (Cat. No. 00CB37048)*. IEEE, 83–90.

[14] Hyunjoo Lee, Jiyeon Lee, Daejun Kim, Suman Jana, Insik Shin, and Sooel Son. 2021. AdCube: WebVR Ad Fraud and Practical Confinement of Third-Party Ads. In *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, 2543–2560. https://www.usenix.org/conference/usenixsecurity21/presentation/lee-hyunjoo

[15] Brigette Lundeen and Jim Alves-Foss. 2012. Practical clickjacking with BeEF. In *2012 IEEE Conference on Technologies for Homeland Security (HST)*. 614–619. https://doi.org/10.1109/THS.2012.6459919

[16] Yoky Matsuoka, Sonya J Allin, and Roberta L Klatzky. 2002. The tolerance for visual feedback distortions in a virtual environment. *Physiology & behavior* 77, 4-5 (2002), 651–655.

[17] Michael J McGuffin and Ravin Balakrishnan. 2005. Fitts' law and expanding targets: Experimental studies and designs for user interfaces. *ACM Transactions on Computer-Human Interaction (TOCHI)* 12, 4 (2005), 388–422.

[18] David E Meyer, Richard A Abrams, Sylvan Kornblum, Charles E Wright, and JE Keith Smith. 1988. Optimality in human motor performance: ideal control of rapid aimed movements. *Psychological review* 95, 3 (1988), 340.

[19] Muhammad Usman Rafique and S Cheung Sen-ching. 2020. Tracking Attacks on Virtual Reality Systems. *IEEE Consumer Electronics Magazine* 9, 2 (2020), 41–46.

[20] Sharif Razzaque, Zachariah Kohn, and Mary C Whitton. 2005. *Redirected walking*. Citeseer.

[21] Gustav Rydstedt, Elie Bursztein, Dan Boneh, and Collin Jackson. 2010. Busting frame busting: a study of clickjacking vulnerabilities on popular sites a survey of frame busting. In *Web*, Vol. 20. 1–13.

[22] Majed Samad, Elia Gatti, Anne Hermes, Hrvoje Benko, and Cesare Parise. 2019. Pseudo-haptic weight: Changing the perceived weight of virtual objects by manipulating control-display ratio. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–13.

[23] Rakhi Sinha, Dolly Uppal, Dharmendra Singh, and Rakesh Rathi. 2014. Clickjacking: Existing defenses and some novel approaches. In *2014 International Conference on Signal Propagation and Computer Technology (ICSPCT 2014)*. 396–401. https://doi.org/10.1109/ICSPCT.2014.6884934

[24] Frank Steinicke, Gerd Bruder, Jason Jerald, Harald Frenz, and Markus Lappe. 2009. Estimation of detection thresholds for redirected walking techniques. *IEEE transactions on visualization and computer graphics* 16, 1 (2009), 17–27.

[25] Pascale Touzalin-Chretien, Solange Ehrler, and André Dufour. 2009. Dominance of Vision over Proprioception on Motor Programming: Evidence from ERP. *Cerebral Cortex* 20, 8 (12 2009), 2007–2016. https://doi.org/10.1093/cercor/bhp271 arXiv:https://academic.oup.com/cercor/article-pdf/20/8/2007/1136398/bhp271.pdf

[26] Ananya Yarramreddy, Peter Gromkowski, and Ibrahim Baggili. 2018. Forensic analysis of immersive virtual reality social applications: A primary account. In *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 186–196.

[27] Zhen Yu, Hai-Ning Liang, Charles Fleming, and Ka Lok Man. 2016. An exploration of usable authentication mechanisms for virtual reality systems. In *2016 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*. IEEE, 458–460.