What's the Thumb Doing? Improving Precision for Thumb-to-Finger Interactions on Hand Proximate User Interfaces

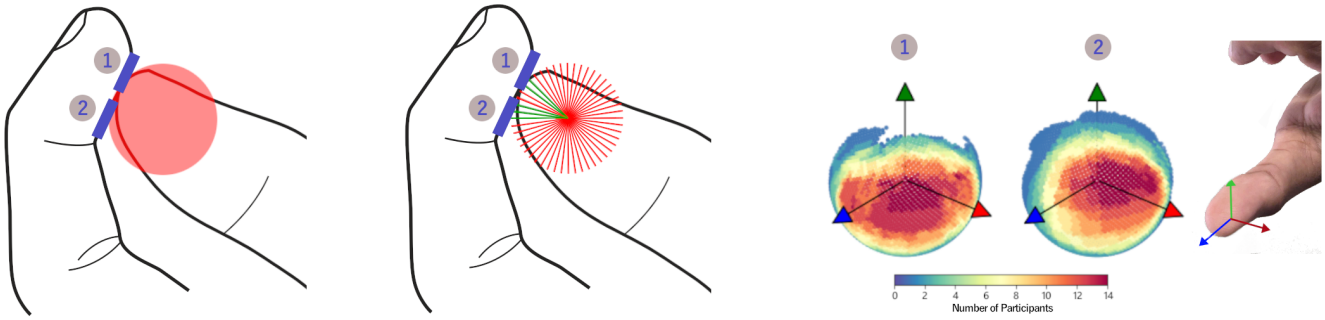
Sharif AM Faleel
amshamoh@student.ubc.ca
University of British Columbia
Kelowna, Canada

Rishav Banerjee
rishavb@student.ubc.ca
University of British Columbia
Kelowna, Canada

Omang Baheti
obaheti@student.ubc.ca
University of British Columbia
Kelowna, Canada

Khalad Hasan
khalad.hasan@ubc.ca
University of British Columbia
Kelowna, Canada

Pourang Irani
pourang.irani@ubc.ca
University of British Columbia
Kelowna, Canada



(a) (b) (c)

Figure 1: Shows the interaction of thumb-to-finger interactions with two targets on the distal phalanx of the index finger on HPUI. (a) Shows using the collision volume that represents the thumb in a physic engine. In this example, the collision volume of the thumb is colliding with both targets. (b) To better understand what happens, the spherical collision volume is replaced with raycasts. The figure shows the cross-section of the raycasts and highlights the rays that are interacting with the targets in green. This raycast-based approach is also further used to improve the accuracy of target selection. (c) Shows the number of participants who had utilized a given ray when selecting the respective targets during the data collection study.

ABSTRACT

Hand Proximate User Interfaces (HPUI) on Head Mounted Displays (HMD) leverage hand tracking to anchor content on the hand and interact with it using thumb-to-finger interactions. Similar to many other interaction techniques on HMDs, HPUI realizes these interactions by combining simple geometry in game engines. This, in turn, leads to accidental triggers, akin to the "fat-finger problem" on touch screens. To address this, we explore and provide insight into how the thumb's surface interacts when using HPUI by approximating the thumb's surface with a large number of raycasts. We observe that different regions of the thumb are used when interacting with different parts of the hand. The results also highlight the need to consider the temporal component. We then propose approaches to improving the precision of thumb-to-finger interactions on HPUI and show that these improve target selection accuracy with denser target layouts.

CCS CONCEPTS

• **Human-centered computing** → *Virtual reality*; *Graphical user interfaces*; **Empirical studies in HCI**.

KEYWORDS

virtual reality, game engines, fat-finger, hand proximate user interfaces, thumb-to-finger interactions

ACM Reference Format:
Sharif AM Faleel, Rishav Banerjee, Omang Baheti, Khalad Hasan, and Pourang Irani. 2025. What's the Thumb Doing? Improving Precision for Thumb-to-Finger Interactions on Hand Proximate User Interfaces. In *Proceedings of*

GI '25, May 26–29, 2025, Kelowna, BC

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

Graphics Interface 2025 (GI '25). ACM, New York, NY, USA, 14 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Head-mounted displays (HMDs) have gained attention as a tool for everyday productivity and a potential smartphone replacement. Hand Proximate User Interface (HPUI) has been proposed as a promising solution to translating smartphone interactions to the interaction space of HMDs [19]. These are interfaces where interactive elements are displayed on and around the hand and thumb-to-finger gestures are used to interact with them. Because of the visual component, they do not suffer from the lack of discoverability like gestural or voice interactions- one simply has to look at their hand and tap with their thumb. While HPUI affords tactility, comfort, social acceptability, and proprioception with extended usage [23], it suffers from accuracy issues [21]. Similar to the fat-finger problem on touch screens [31, 55], target selection with the thumb is imprecise even when the comfortable interaction space primarily resides on the surfaces of the other fingers.

Augmenting the hand with specialized hardware or using gloves would be a potential solution for realizing thumb-to-finger interactions on HPUI [7, 11, 36]. However, in practice, such augmentations would be limited in their practical use, particularly when considering HMDs as a day-to-day computing platform. Imagine following a tutorial video while cooking and interacting with video controls on HPUI - a glove with capacitive elements would not be a practical choice here. As a result, hand pose estimation has become a common approach for hand-based interaction [45]. As with many other hand-based interactions on HMDs, HPUI, and similar interaction techniques are also realized through physics- or game-engines [24, 48, 60]. Hence, we focus on using such physics engines for HPUI interactions.

Now, consider tapping the tip of the index finger and tapping on the proximal phalanx of the little finger. The part of the thumb that makes contact in these two instances would be different. Hence, the geometry that represents the thumb would have to be large enough to account for this. When interactive elements are placed next to each other, which would be necessary to take advantage of the interactive space on the fingers, the larger geometry of the thumb results in more targets getting accidentally selected, as seen in Figure 1 - the fat finger problem. To put it differently, the size of the thumb's collision geometry relative to the collision geometry of any targets on the fingers is relatively large. There are two components we can analyze to better understand and optimize the interaction on HPUI. First is the elements on the fingers the thumb would interact with. Following the convention used by Unity Extended Reality Interaction (XRI) Toolkit, we refer to these as *interactables* [1]. However, these interactables can take many forms - they can be a large continuous surface spanning multiple fingers [48], or small elements only on one part of the finger [43, 58]. That is, unlike the touch screen, it is not one fixed surface. On the other hand, the thumb's collision geometry, referred to as the *interactor* [2], would be the same across a wide range of interactions on HPUI. This would be the second component we could analyze. This is also unlike touch screens, which have much less information on what the finger is doing, while with the hand-tracking-based solutions we use for HPUI, we can closely monitor what the finger is doing. Hence, to better understand what happens during interactions on HPUI, we focus on this second approach - what the thumb does.

We start by collecting data on how users interact with targets on HPUI. In particular, we are focusing on taps on on-finger targets. We use the Unity game engine for our analysis. To get granular detail on what happens during interactions, we use an array of ray casts instead of collisions of simple geometries. The rays are cast in every direction from the center of the interactor on the tip of the thumb (see (b) in Figure 1). We observe that when targets are more densely placed, there is a much higher error rate. Analysis of the ray casts shows that the region of the thumb which is used to interact with different parts of the interaction space varies significantly. Inspired by the touch screen interactions, we propose improvements in detecting taps using thumb-to-finger interactions to reduce false positives. In particular, we use an updated heuristic to rank the interactables and also consider the temporal component of interactions. We finally run a user study that shows the improvements to the accuracy on a denser layout. Our core contributions are two-fold: (1) We provide better insight into how the thumb interacts during thumb-to-finger interactions based on using raycasts to model the thumb in a physics engine. (2) Based on the insights we provide improvements for HPUI that improve the accuracy of target selection on denser layouts.

2 RELATED WORK

2.1 Hand Proximate User Interfaces

Hand Proximate User Interface (HPUI) expands on the literature on micro-gestures [16, 19, 22]. In particular, it expands on thumb-to-finger micro-gestures [33, 52, 54]. In addition to the physical comfort, thumb-to-finger interactions are also seen as more socially comfortable when compared to other micro-gestures [22, 48]. However, a drawback of micro-gestures, and gestures in general, is the lack of discoverability [2, 10, 42]. HPUI addresses this by taking cues from the smartphone interaction and displaying the interactive content directly on and around the hand [19]. In addition, the interactions are similar to the now ubiquitous single-handed mobile usage [35]. However, because of the biomechanics of thumb-to-finger interaction, the display surface is non-contiguous, limited, and changes shape as interactions happen [48]. To circumvent these shortcomings, prior work has explored expanding in the interaction space [3, 42] and also propose designs that take these limitations into consideration [48]. Furthermore, recent work has also shown that HPUI performs on par with other common unencumbered interaction techniques on HMDs [21]. Given the proprioceptive and tactile affordance of interacting with the hand [28, 29], HPUI has also been shown to have better eyes-free affordance while also being easy for novice users to start using the interface [23].

2.2 Detecting interactions on Head Mounted Displays

The broad range of affordances with HMDs has resulted in a wide range of interaction techniques and a broader range of approaches to detect these interactions. When considering mobile and in-situ

[1] link to XRI Interactables
[2] link to XRI Interactors

applications, for which HMDs are well suited, researchers have argued that such interactions must be unobtrusive, non-distracting, comfortable, and efficient to use [17, 18, 26, 44, 53]. Unencumbered interaction techniques are commonly preferred for this reason. When considering hand interactions and micro-gestures in particular, there are commonly two different approaches. First is to augment the hand with sensing technologies, ranging from gloves [9, 34] and rings [13, 61] to using bio-acoustic [40, 62] and bio-impedance [56] sensing. The rings and similar wearables that are designed to be less obtrusive often would only be able to detect a subset of the interaction space of thumb-to-finger interactions such as HPUI. While gloves would allow more precise and broader hand pose estimation, they are often far more obtrusive. The second approach to detecting hand interactions is the use of vision-based hand pose estimation [63]. Its success has seen it being widely adopted by commercial HMDs as well. While it may not be able to detect interactions outside its field of view, its convenience makes it a practical choice for hand-based interactions.

When detecting micro-gestures such as those used with HPUI, the system could provide an output that indicates which gesture was performed. However, this approach would limit the options to the states only supported by the system. Alternatively, using the hand pose estimation and allowing the designers to define how the hand posture is used for different interactions allows for a richer set of interactions. In fact, interaction techniques on HMDs are commonly modeled using this approach. Using a physics engine, such as in Unity, collision volumes or raycast are anchored to a tracked entity like the hand [51], eye-gaze [6], head [49], etc. Following the physical interaction metaphor, interactions are modeled as these tracked entities colliding with interactive virtual surfaces or objects which are also represented by collision volumes. While accurate tracking of the physical movements of the users is a large research topic in its own right, there also has been significant research on understanding [5, 14, 59] and improving the interactions [4, 12, 39] within such physics engine based interactions. Prior work on HPUI and similar interaction techniques also use a similar approach to detecting interactions [19, 24, 50]. Often, the thumb is modeled using a sphere. However, as stated in the introduction, trying to represent the whole thumb surface using a single collision volume, results in incorrect target selection. Prior work that uses this approach for microgestures has seen higher error rates, potentially caused by this [21]. We liken this phenomenon to the fat-finger problem on touch screens. However, the results from touch screens cannot be directly adopted here as the interactive surface is not flat or static. Taking inspiration from the work on understanding how users interact with touch screens [31], our works explore how the thumb interacts with the finger surfaces in HPUI.

3 DATA COLLECTION OF THUMB-TO-FINGER INTERACTIONS ON HPUI

To better understand what may cause the errors, we analyze which part of the thumb's surface interacts with a given target. Insights from this would provide a basis for improving performance and also reducing the error rate with HPUI interactions. The interaction space on HPUI provides for targets on the fingers, between fingers, and off fingers as well [19, 48]. For simplicity, we limit the current analysis only to the volar surface of the fingers.

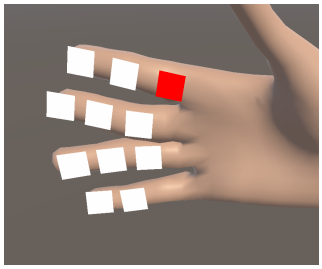


Figure 2: The sparse layout used in the studies.

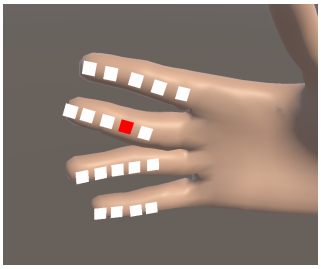


Figure 3: The dense layout used in the studies.

3.1 Targets and tasks

Figure 2 and Figure 3 show the targets and the layout we used for data collection. We used two sizes of targets. Similar to prior work [19], the first is $1cm \times 1cm$ square, which is expected to approximate the finger's width. The second size we used is a $5mm \times 5mm$ square, approximately half the finger's width. The targets were also laid out in two different configurations. With the larger targets, only one was placed on a given phalanx, referred to as the *sparse* layout. The smaller targets were laid out to be more dense, with two targets on a phalanx, referred to as the *dense* layout. This allows us to see the impact of the density of displayed targets. In all cases, the target on a given phalanx was anchored to the adjoining joints of a given phalanx. That is, targets on the distal phalanx were anchored relative to the tip and distal interphalangeal joint of the finger, and the targets on the intermediate phalanx were anchored relative to the distal interphalangeal joint and proximal interphalangeal joint. We excluded the location closest to the proximal phalanx for the index, middle, and ring finger with the dense layout, and avoided placing targets on the proximal phalanx of the little finger in both layouts. As noted by prior studies [19, 48] as well as during our pilots, some participants struggle to reach these regions with their thumb during thumb-to-finger interaction and are generally uncomfortable to interact with [33, 41]. Further, we are only considering targets on the volar side of the fingers. This resulted in 11 targets on the sparse condition and 19 targets on the dense condition.

The location of the interactor was on the distal phalanx of the thumb. It was placed 90% along the segment from the distal joint to the thumb tip. The ray casts were made to be spaced equally originating from the location of the interactor. To determine the ray directions, points were placed on a sphere aligned to the thumb's distal phalanx's orientation using the Fibonacci grid approach. Distributing points on a spherical surface is a well-known mathematical problem [30]. The Fibonacci grid is a well-known solution widely used for various applications. Using an approach like latitude-longitude coordinates would result in more points clustered at the poles [25]. We placed 5184 points, which is the same number of points that would result from using latitude-longitude coordinates with points spaced 5° apart along both latitude and longitude axes. A ray was cast from the location of the interactor to each point, resulting in 5184 rays.

For this analysis, we consider the following axes on the thumb (also seen in Figure 5):

- **Y**: orthogonal to the distal phalanges thumb surface.
- **Z**: Along the joint - from the IP joint to the tip of the thumb.
- **X**: Right from the thumb when looking at the back of the thumb. This would be the cross-product of the Y and Z axes.

The collision geometry of each target is made to match the visuals of the targets. Throughout the study, we record all the rays that make contact with any of the target interactables. A selection of a target happens when any of the rays intersect any of the colliders representing a target and the distance of the ray is below a threshold. We refer to this threshold as the *selection distance threshold*. In this study, it was set to 20mm. For each ray, when it collides with any of the targets, we record the distance from the origin of the ray, which is the position of the interactor itself to the point of collision, and which target it made contact with. In a given frame, when more than one target is within the distance threshold, the one with the shortest ray is treated as the target being selected. No other targets would be selected until the tap gesture ends, i.e., when no targets are within the selection distance threshold for any of the rays.

We use a simple sequential tapping task. The participants were expected to select the target highlighted in red. All other targets would be white. When any target gets selected, it flashes green and an audio cue is played. If the participant maintains contact with the target, it will remain a lighter green hue. A trial in this data collection study is when the participant successfully selects the highlighted target. However, the trial progresses only when the tap gesture ends. This was done to ensure data of the complete tap gesture is recorded.

A half-start approach was used during the data collection study, where one participant starts with the dense condition and another starts with the sparse condition. Participants completed one condition before they moved to the next one. In the sparse condition, which had larger targets, each target was selected 20 times, in the dense condition with the smaller targets, each was selected 10 times. The sequence of targets presented to the participant was randomized such that the same target was not presented in two consecutive trials.

Figure 4: The setup used for the studies.

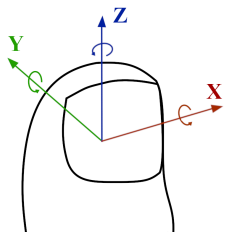


Figure 5: The plane of the (right) thumb and axis used during the study Section 3

3.2 Apparatus

Computer vision-based hand-tracking solutions still lack the fidelity necessary to test the granular thumb-to-finger interaction used in our study. However, we expect the ongoing efforts on improving hand-tracking using vision-based approaches to achieve better real-time performance to enable using such interactions. Hence, we use a Vicon motion tracking system with 13 cameras for hand-tracking as seen in Figure 4. The input from the 13 cameras are synthesized via the Vicon Nexus 2.10.3 software and streamed via a custom Python web-socket implementation for real-time tracking. We also track the position of the HMD with the same tracking solution so that the hand and the HMD share the same coordinate space. For the studies, we use the Meta Quest 3 HMD. The experiment application is implemented in Unity and was executed using the Meta Quest Link. This was done to avoid any performance penalties. We also used the Unity Experiment Framework to manage the experiment [8].

3.3 Participants

The study was approved by the Research Ethics Board of local universities. Data was collected from 14 participants (10 male, 4 female). The average age of the participants was 26.4 ($SD = 3.89$). 12 of the participants were right-handed, and 2 were left-handed. All participants have had some experience using HMDs. They had rated their experience on a 7-point Likert scale, with 1 being "no experience" and 7 being "A lot". The median score was 3.5 ($M = 3.57, SD = 1.69$). We also asked participants to rate their experience with hand tracking on the same 7-point Likert scale. The median score was 2.5, with two participants reporting a score of 1.

3.4 Procedure

When a participant arrived and completed the consent form, they were asked to complete a brief demographics questionnaire. Following an explanation of the procedure, the participants are asked to wear the marker glove on top of a disposable glove. They are asked to be seated inside the capture volume of the Vicon motion capture system, where we ensure the tracking software is calibrated to their hands. They were then asked to wear the VR headset. The participants were asked to be as "fast and accurate" as possible. They are first presented with a practice block which had the same number of trials as the first block to make the participant familiar with the interaction technique and the experiment setup. They are then administered the trials of the study. To avoid fatigue, the trials are separated into four blocks, two blocks for each condition. The participants are provided a break between each block.

3.5 Analysis

3.5.1 Overall Performance. We first report the overall performance of the observed data. Note that we primarily report descriptive statistics here for completeness, as the focus is on what happens during the interactions. We first compute the error rate. In our analysis, we refer to error rate as the ratio between the number of errors made by a participant for a given condition and the number of correct selections made:

$$error\ rate = \frac{\#\ of\ errors}{\#\ selections}$$

Figure 7a shows the summary of the error rate. The dense layout had a mean error rate of 0.41 ($SD = 0.17$) compared to the 0.11 ($SD = 0.08$) for the sparse layout. Figure 7a shows the summary of the selection time, i.e., the time to make the correct selection once the target has been highlighted. Here, the dense layout had a mean

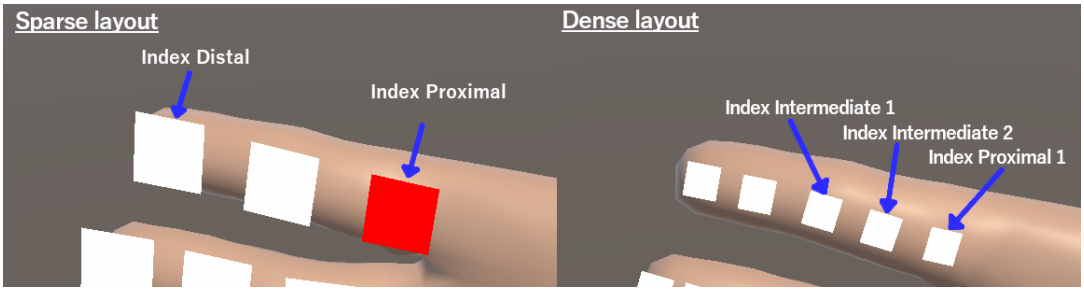
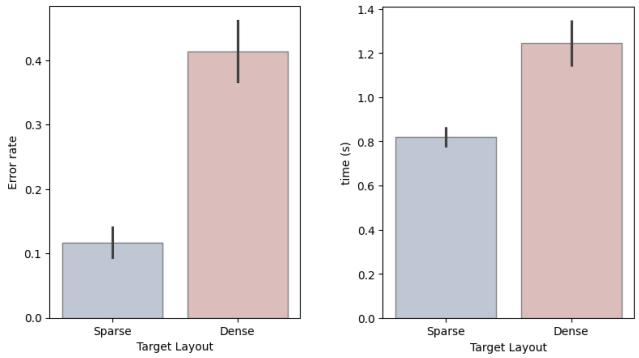


Figure 6: The example targets referred to in Section 3.5 and Figure 13.

selection time of 1.24s ($SD = 0.36s$) and the dense layout had a mean selection time of 0.81s ($SD = 0.15s$).



(a) The error rate of the two different layouts.

(b) The selection time of the two different layouts.

Figure 7: The overall performance of the data collected. Shows standard error bars.

3.5.2 Analysis of Regions. We first explore what happens on the thumb when the thumb is in contact with the interactable. In this section, we primarily focus on the tap gesture that resulted in the correct selection of the highlighted target. Further, for this analysis, we are considering the frame that had the largest number of rays making contact with the intended interactable. This gives us the most amount of information on what happened when the target was selected. For each gesture frame, of the 5184 rays that are cast, we look at which of them made contact with the intended interactable. To better visualize the data, we mirrored the results from the left-hand participants around the Z-axis (see Figure 5). Figure 8 shows the number of participants who used a given ray when selecting the respective target for each target in the sparse layout. Figure 9 shows the same but for the targets on the index finger for both dense and sparse layouts. Each row shows the rays used for each phalanx of the index finger.

Figure 9 shows that the rays used for the dense targets are generally less compared to selecting targets in the sparse condition. At the same time, the rays used for selecting targets on a given phalanx during the dense condition are also a subset of the rays used for the larger targets in the sparse condition on the same phalanx. Another observation we make is that the part of the thumb that makes contact also greatly differs. Overall, when selecting a target placed on the volar side of a finger, the ulnar side of the thumb is being used the most. When interacting with the distal phalanx of any finger, we see that there is a higher variance in which part of the thumb is being used to make a selection. Whereas when the targets are closer to the palm, the region is more focused. This could be explained by the biomechanics of the thumb-to-finger interactions, where with the distal phalanx, as all three joints of the finger are involved, there are more degrees of freedom. Which results in more directions in which the thumb can connect with the tip of a finger. However, with the proximal phalanx, for example, there is only one joint. The lesser degrees of freedom and the lesser reach of the thumb result in fewer directions or ways the thumb could connect with a target placed on that phalanx.

3.5.3 Analysis of Gestures. In this section, we explore the possible causes of the higher number of errors seen in Section 3.5.1. In the data collection study, the interaction happens at the very beginning of the gesture. That is, the decision of which target to select is made in the very beginning based on the distance - where the interactable with the shortest distance is selected. This could be one reason for the higher number of errors. Figure 10, shows the distance of the ray that recorded the shortest distance change over time. Note that, the distance of a ray in our analysis is the distance from the origin of the ray to the point when the ray makes contact with an interactables collision volume. Similarly, Figure 11 shows the change in the number of rays. We see that throughout the gesture the shortest distance recorded follows the "U" shape, and the number of rays has an up-side-down "U" shape. Presumably, using the information when the shortest distance or the highest number of rays from the whole gesture could result in better performance. To assess this, we extracted the trials that had recorded any number of errors. This resulted in 1051 trials. That is, in these trials, more than one tap gesture was performed to correctly select the intended target. If the intended target had been selected on the first tap gesture, it would not have been counted as an error. We then consider the shortest distance during the entire tap gesture or the highest number of rays during the entire tap gesture to decide which interactable should be selected. Note that, the results in Section 3.5.1 are based on the distance at the beginning of the gesture. That is, the decision of which target should be selected was made in the first frame of the gesture. We analyze if using these metrics, the shortest distance or the number of rays from throughout the gesture, would have resulted in the intended

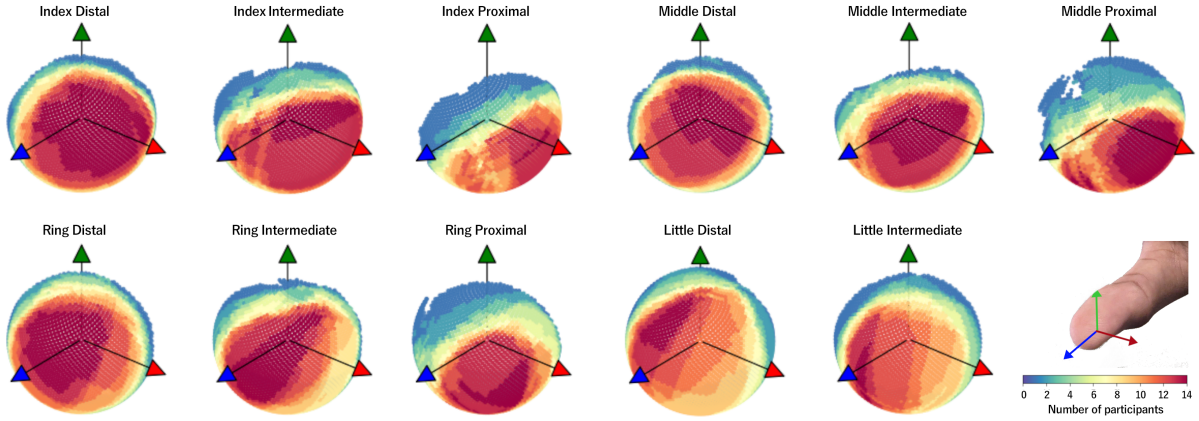


Figure 8: Each plot shows the rays used when in contact with each target in the sparse condition. For each target, the plot shows the number of participants who used a given ray when selecting the respective target. The rays are from the frame where the highest number of rays were in contact with the intended target during a given tap gesture.

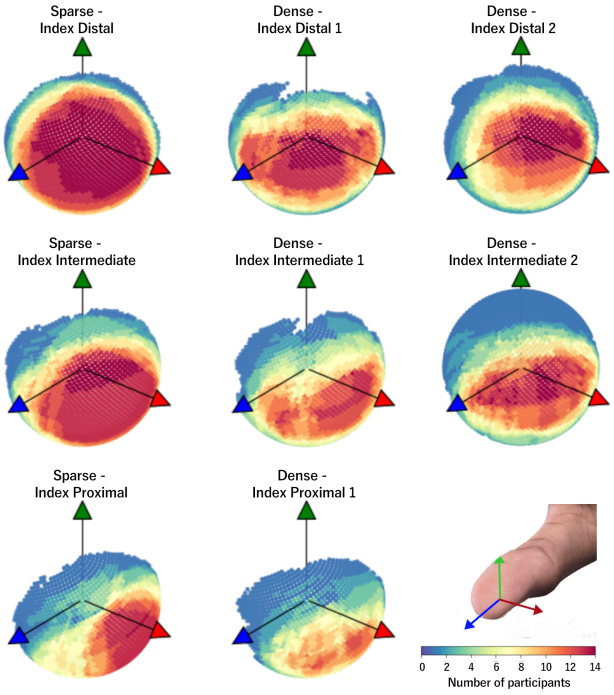


Figure 9: Each plot shows the rays used for the target on the index finger. For each target, the plot shows the number of participants who used a given ray when selecting the respective target. In each row, the first is the target in the sparse condition. The remainder corresponds to the targets in the dense layout. The rays are from the frame where the highest number of rays were in contact with the intended target during a given tap gesture.

targets being selected in the first tap gesture. Figure 13 shows one trial where the correct target was selected in the second tap. In 69.9% of these 1051 trials, the intended target was selected in the first tap gesture if the shortest distance from the whole gesture is considered. When the number of rays was used, it showed that 74% of the trials would not have had errors. In 6.1% of the trials, the intended target had recorded the highest number of rays but not the lowest distance in the first tap gesture. In 2% of the trials, the intended target had recorded the lowest distance but not the highest number of rays in the first tap gesture. Figure 13 shows an example where the first gesture has recorded the highest number of rays in the first gesture, but not the lowest distance. This implies that using this information of the shortest distance and the largest number of rays in the entire gesture would be a better indicator of which target is to be selected. This also further elucidates the analysis in Section 3.5.2, where we consider the frame with the largest number of rays interacting with an intended target.

Another observation we made during this analysis is that the shortest distance is much lower than the 20 mm threshold we used during the data collection. The Figure 14 shows the largest of the mean distance of a given ray for each participant. Similar to the analysis in Section 3.5.2, here also we are considering only the frame with the largest number of rays making contact with the intended interactable. We consider the largest value as opposed to taking the mean as different users have different finger thicknesses. Having a shorter distance could disadvantage users who have larger fingers. While this is a factor that can impact performance, this work focuses on building a broader understanding of the interactions and defers personalization and its advantages to future work. The mean of these largest distances was 13.8 mm. Similar to the above analysis, we also looked at how many of the trials with errors would have resulted in no errors if the same selection process used when collecting data is used but with the shorter distance threshold. In this case, 13.9% of trials would have resulted in the first tap gesture selecting the intended target.

Another component we considered was the time to select a target. The Figure 12 shows the histogram of the time it took to make the first erroneous selection since the trial started. We can see that

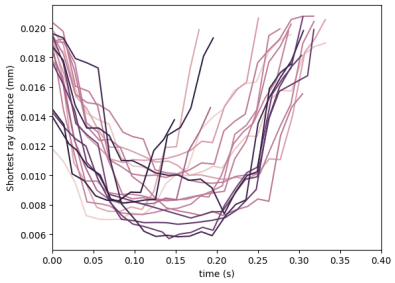


Figure 10: Shortest ray distance when a ray is interacting with the intended target plotted over time for sample tap gestures.

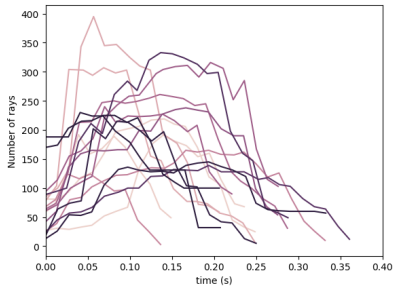


Figure 11: Number of rays in contact with the intended target plotted over time for sample tap gestures.

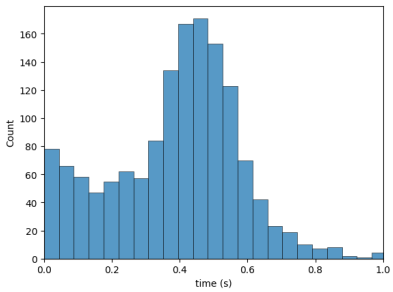


Figure 12: Histogram of time it took to make the first erroneous selection since the trial started.

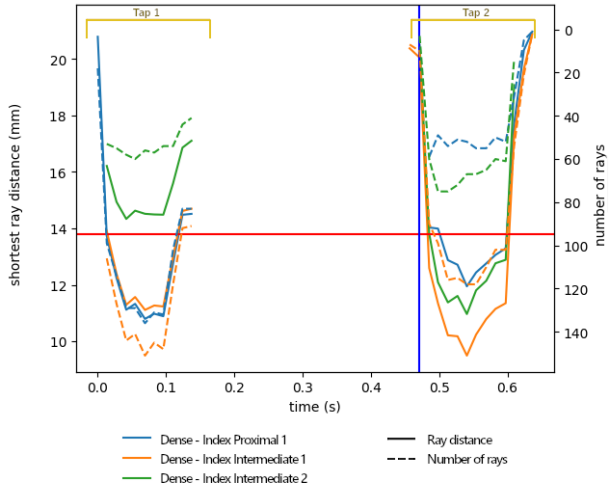


Figure 13: An example trial with the dense condition where the intended target was *Index Proximal 1* (see Figure 6). Shows the distance of the shortest ray and the number of rays over time. The trial had one error (Tap 1), where Index Intermediate 1 was selected. Tap 2 is the gesture that resulted in the correct target being selected. The red horizontal line is the 13.8 mm threshold computed in this section. The blue vertical line was when the tap event was triggered during the data collection.

selection had happened as soon as the trial had started. This happens mostly as a result of the next target being right next to the selected target. As the participant moves away, the selection ends, thereby starting the next trial. The adjacent target immediately gets selected as the trial starts. Hence, in addition to using the information of the whole gesture, a possible debounce threshold also needs to be considered to reduce such errors.

We also considered the time it takes to complete a gesture. Figure 15 shows the time of the tap gesture when the correct target was selected with the 20 mm threshold used for the data collection and the 13.8 mm threshold computed above. As expected, the time of the tap gesture decreases with the distance.

Additional analysis of the data can be seen in Appendix A. In Section 4, we consider all of these observations to propose improvements to how a target is selected.

4 IMPROVEMENTS FOR TAP DETECTION FOR HPUI

Based on the analysis in Section 3.5, we consider two improvements to the detection. The detection mechanism can be decomposed into two components. The first is detecting and ranking the interactables within the selection radius during a given frame. The second is to consider the temporal component. That is given a sequence of frames, it would manage the states and decide which interactable would receive the corresponding event. We propose improvements to both of these components here.

4.1 Improving the Ranking Mechanism

During the data collection study in Section 3, a target was determined based on the shortest ray. That is, the shortest ray distance of a given interactable was used as a score, and the interactable with the lowest score ranked at the top. Here we include the number of rays as another component of the score used to rank the interactables. For the interactable i during the frame j, when $d^{ij}_{shortest\_ray}$ is the distance of the shortest ray as used in Section 3 and n^{ij}_{rays} is the number of rays that interacted with the collider of the interactable i and N^{j}_{rays} is the total number of rays interacted with any interactable in frame j, we express the score as a ratio between the distance and the normalized number of rays:

$$score^{ij} = \frac{d^{ij}_{shortest\_ray}}{\frac{n^{ij}_{rays}}{N^{j}_{rays}}} \quad (1)$$

4.2 Improving State Management

The results from the data collection clearly show that using the first frame is not ideal, and can result in errors. Taking inspiration from the literature on three-state virtual keyboards [20, 37], we propose a delayed activation. In other words, the gesture is triggered on "finger-up" instead of "finger-down". The algorithm used for this is outlined in algorithm 1. In simpler terms, when a new gesture starts, this algorithm tracks all the interactables that get selected (line 16). When the gesture ends (line 3), if the time of the gesture

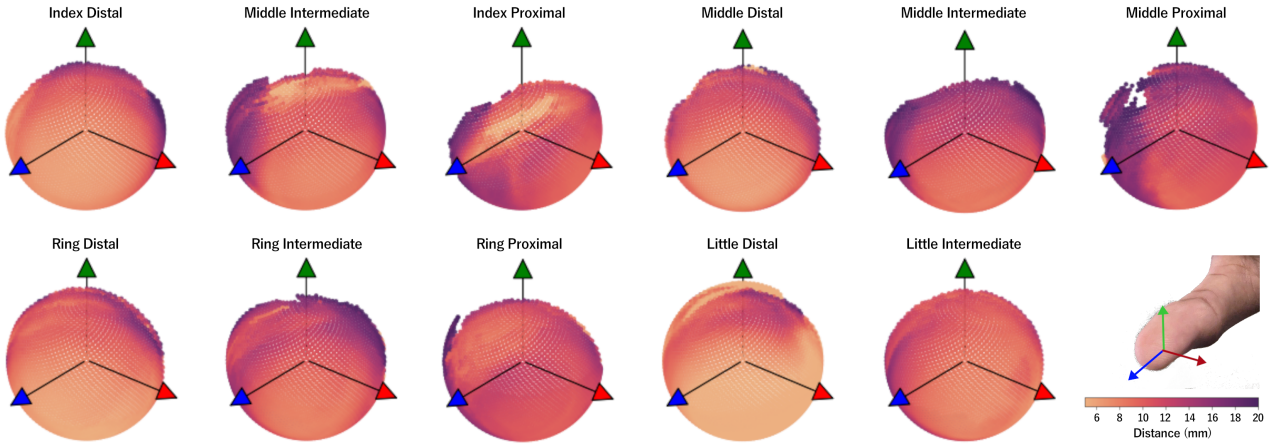


Figure 14: Each plot shows the computed distances of the rays used when in contact with each target in the sparse condition. The rays are from the frame where the highest number of rays were in contact with the intended target during a given tap gesture.

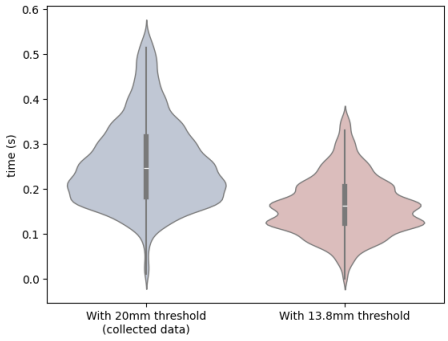


Figure 15: Violin plot of the time taken to select a target during the data collection (with 20 mm threshold), and the time it would have taken if the 13.8 mm computed threshold was used.

Algorithm 1: Tap algorithm for HPUI

Input : $I_{current}$ /* Interactables selected in current frame */
/* Data tracked across frames */
Tracked : $I_{tracked}$ /* Tracked interactables */
Tracked : $F_{tracked}$ /* Tracked frame indices */

1 $t_{current} \leftarrow$ current time;
2 **if** $I_{current} = \emptyset$ **then**
3 **if** $I_{tracked} \neq \emptyset$ **then**
4 $t_{gesture} = t_{current} - t_{start}$;
5 **if** $t_{gesture} > T_{debounce}$ & $t_{gesture} <= T_{tap}$ **then**
6 $i_{target} = i \in I_{tracked}$ such that $score^{ij} < score^{xj}\ \forall x \in I_{tracked}\ \forall j \in F_{tracked}$;
7 Trigger tap on i_{target};
8 **end**
9 $I_{tracked} \leftarrow \emptyset$;
10 $F_{tracked} \leftarrow \emptyset$;
11 **end**
12 **else**
13 **if** $I_{tracked} = \emptyset$ **then**
14 $t_{start} \leftarrow t_{current}$;
15 **end**
16 $I_{tracked} \leftarrow I_{current} \cup I_{tracked}$;
17 $F_{tracked} \leftarrow F_{tracked} \cup$ current frame index;
18 **end**

(line 4) is between the debounce threshold ($T_{debounce}$) and tap time threshold (T_{tap}), it triggers the tap event (line 7). The debounce threshold eliminates accidental tap triggers seen in Figure 12. We include a tap time threshold to prevent users from pressing down and micro-adjusting to select the correct target. Note that our focus is on tapping behavior. However, such complex gestures are out of the scope of the current exploration. This requires that $T_{debounce}$ is smaller than T_{tap}. The interactable that receives the tap event would be the interactable that had the lowest score in the time window of the gesture (line 6).

5 COMPARING IMPROVED TAP DETECTION

We executed a follow-up user study to validate the results of the improvements made to the tap detection algorithm.

5.1 Task and Study Design

We use the same task we used in Section 3 with the same layout and sizes of targets. However, we compare the following factors:

- *Score:* Distance only score vs Improved score (Section 4.1).
- *State management:* Simple gesture (where $T_{debounce} = 0$ and $T_{tap} = \infty$) similar to Section 3 vs Improved tap detection (Section 4.2)
- *Layout:* Same as Section 3; sparse vs dense.

For the improved tap detection, we use 50ms for $T_{debounce}$ and 300ms for T_{tap}. Also, the target selection distance threshold was set to 13.8mm, which is the 90^{th} percentile of distances used by all the rays used in any gesture from the data collection study (see Section 3.5). The 90th percentile was chosen based on trial and error, where any lower distances had higher false negatives. Note that the combination of *distance only* and *simple gesture* would function as the baseline, with the combination of *improved score* and *improved tap gesture* would be the combined solution we are comparing.

We counterbalanced on the *score* and *state management* factors. This way the participants would complete both layouts for each

combination of the above two factors. However, half the participants did the dense layout first for all four combinations, while the other half did the sparse first. To avoid fatigue from playing a role, each target on the sparse layout was selected 6 times and each target on the dense layout was selected 3 times.

The procedure and the apparatus were the same as the data collection study in Section 3.

5.2 Participants

Here also, the study had been approved by the local university's Research Ethics Board. Data was collected from 18 participants (11 male, 7 female). The average age of the participants was 24.2 ($SD = 3.39$). 17 of the participants were right-handed, and 1 was left-handed. They had rated their experience with HMDs on a 7-point Likert scale, with 1 being "no experience" and 7 being "a lot". The median score was 2 ($M = 2.5, SD = 1.61$) with 5 rating their experience with a score of 1. We also asked participants to rate their experience with hand tracking on the same 7-point Likert scale. The median score was 1 ($M = 1.7, SD = 1.2$), with 8 participants reporting a score of 1. We analyzed the correlation of the metrics analyzed below with the responses from the survey but found no interesting patterns, so these results are not included for brevity.

5.3 Results

5.3.1 Error rate. Similar to Section 3.5, here also we analyze the error rate. For the analysis, we combined the score and state management factors into one factor we refer to as *condition*. As the data violated the assumptions for RM-ANOVA, we analyzed the data with Friedman's test and used Kendall's W to compute the effect size[3]. We analyzed the data for the dense layout and sparse layout separately. With the sparse layout, the condition did not have a significant effect on the error rate ($\chi^2 3 = 2.277, p = 0.43$, small effect size Kendall's W = 0.05). For the large targets, the combined approach (*improved score + improved tap*) reports the lowest mean error rate ($M = 0.085, SD = 0.05$). Interestingly, the improved score (*improved score + simple-gesture* ($M = 0.11, SD = 0.05$)) and improved tap (*distance only + improved tap* ($M = 0.11, SD = 0.10$)) had a slightly worse error rate than the baseline, i.e. *distance only + simple-gesture* ($M = 0.09, SD = 0.03$). Whereas with the dense layout, the condition had a main effect ($\chi^2 3 = 9.42, p = 0.02$, small effect size Kendall's W = 0.17). We then conducted a post-hoc analysis with Bonferroni correction for the dense layout. The baseline ($M = 0.32, SD = 0.1$) had the highest error rate among the conditions and the combined condition had the lowest error rate ($M = 0.23, SD = 0.11$). The only significance we observe in the post-hoc analysis of the dense layout is also between the baseline condition and the combined approach. In other words, the improved score (*improved score + simple-gesture* ($M = 0.26, SD = 0.09$)) or improved tap (*distance only + improved tap* ($M = 0.28, SD = 0.15$)) on their own reduced the error rate, but did not show a significant improvement, only when combined we observe a significant improvement to the error rates. The summary of the results can be seen in Figure 16.

[3]For the interpretation of effect sizes, Cohen's guideline was followed [15]

5.3.2 Selection time. For completeness, we also analyzed the time to make a selection. The selection time is computed as the time for the selection event to be triggered by the user after the target is highlighted. Similar to the error rate, the selection time also violated the assumptions of RM-ANOVA. Hence, here also we conducted Friedman's test and use Kendall's W for effect size. Here also we analyze the two layouts separately. The condition did not have a main effect with either sparse layout ($\chi^2 3 = 4.38, p = 0.22$, small effect size Kendall's W = 0.07) or dense layout ($\chi^2 3 = 1.5, p = 0.68$, small effect size Kendall's W = 0.025). With the dense layout, using only the improved score shows the lowest time ($M = 1.02s, SD = 0.12s$) followed by the baseline ($M = 1.05s, SD = 0.14s$). Similarly, with the sparse layout, the lowest time is seen with the baseline ($M = 0.79s, SD = 0.07s$) followed by the condition where only the improved score was used ($M = 0.81s, SD = 0.06s$). That is, even though not significantly different, the use of the improved tap slightly increases the time to select. This could be explained by how the selection time is computed - with the simple-gesture, the event is triggered at the very beginning of the tap gesture. Whereas with the improved-tap, the event is triggered when the gesture is completed. Figure 17 shows the summary of the results. Though not significant, the *improved tap* increases the selection time.

6 DISCUSSION

6.1 Exploration of Factors Influencing Accuracy in HPUI

The aim of our work was to understand what happens during interaction with HPUI. In addition to providing insight into what happens as interactions take place, the raycasts-based approach also provides mechanisms to further improve the selection accuracy. However, contrary to our expectations based on the analysis of the gestures (Section 3.5.3), we did not observe a significant improvement with the larger target sizes. Though it is worth noting that the error rate observed with the proposed improvements (8.5%) is much lower than the error rate reported by Faleel et al. [21] (19%). But this could potentially be a result of the different tasks employed, where Faleel et al. [21] used a transcription-like task while we used a simpler target selection task. Whereas Huang et al. [33] show significantly lower error rates even on layouts similar to the dense layout we have used. Here the difference could be a result of the approach they used to detect interactions - where they used a more controlled setup for detection which may not generalize.

With the dense layout of targets, we see a significant improvement in the error rates. But only when the proposed improvements are combined. The error rate of 28% with the dense targets would still be relatively high in practice. Further research is required to enable the selection of such denser layouts with the hand pose estimation-based approach. Hand pose estimation is a rapidly evolving area of research separate from our current investigation [1, 45], which is why we opted to use a motion-tracking system instead. However, ongoing advancements in hand pose estimation could enhance the applicability of our exploration by enabling more complex and accurate interactions. We had only considered the temporal component (see Section 4.2) and improving the heuristic (see Section 4.1). While we did not directly compare the impact of different distances, we see that there error rates for both types of targets are much lower in the

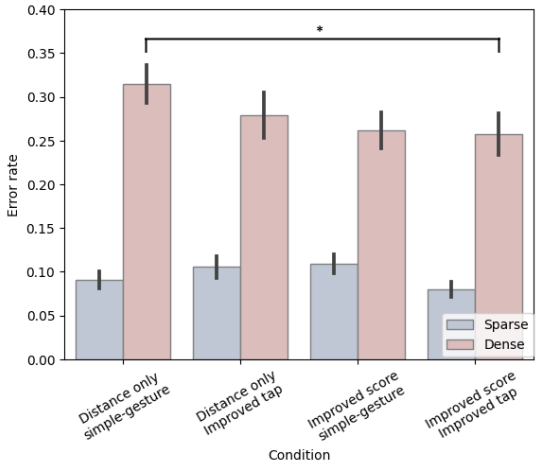


Figure 16: Summary of the error rate results. Shows the standard error bars. In addition to the significance shown, all Sparse layout conditions were significantly different from all Dense layout conditions.

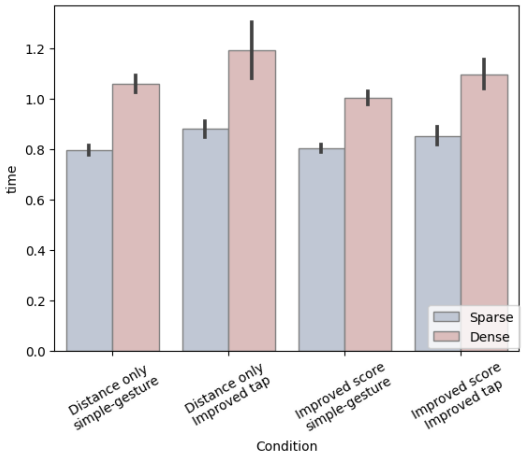


Figure 17: Summary of the time results. Shows standard error bars.

follow-up study in Section 5 when compared to the data collection study Section 3. In Section 3 a selection distance threshold of 20 mm was used, whereas in Section 5 it was set to 13.8mm. However, there are other factors we had not explored that could further improve the performance of the target selection. Our analysis emphasizes a few other factors:

- The use of different distances for each ray to emulate a shape different from a sphere (see Figure 14).
- Using dynamic information such as the trajectory of the thumb relative to the interactables to choose a different set of rays (see Figure 20) or heuristics.
- Personalizing the values based on the user characteristics (see Figure 18 and Figure 19).

Future work would further explore these factors and how they can be used to improve the accuracy of selecting smaller denser target layouts.

The ranking heuristic we propose (Section 4.1) can also be understood as having a better approximation of the area of contact and distance to improve the accuracy of selections. Most interaction techniques in HMDs that use physics engines often use the distance as a primary metric when making decisions when multiple possible targets can be selected [4, 39, 51]. In addition to the distance, we also consider the number of rays. The number of rays on its own only provides the angular area, or solid angle, as the rays are spaced equally. Whereas the heuristic in Equation 1 can be seen as prioritizing the target closest to the interactor with the largest angular area. Finally, to ensure our findings are accessible by researchers and developers, we have integrated our results with the open-source HPUI-Core framework [4].

[4]HPUI-Core on Github

6.2 Comparison to Solutions from Related Interaction Techniques

In this section, we situate the proposed techniques within the related literature. Thumb-to-finger interactions on HPUI share similarities with touch screen interactions, such as the fat-finger phenomenon, where finger occlusion and touch ambiguity reduce accuracy. On touch screens, researchers have shown that the posture of the finger relative to the screen has an impact on the accuracy [31, 32]. However, these insights cannot be directly translated to HPUI because the interactions are not on a static two-dimensional surface. In addition, as with many other interaction techniques on HMDs, interactive elements on HPUI can take many shapes [24, 48] and sizes [48, 50, 60]. Irrespective of the differences, the interactions are still taps. In fact, algorithm 1 in particular is similar to some of the related work that explores interactions on touch screens [20, 37, 38]. They also share similarities with solutions for mid-air target selection in VR and AR [39, 46], addressing issues like human motor inaccuracies and unintended interactions [27, 57, 59]. Future work can further expand on the similarities to address the fat-finger problem on touch screens. Our approach to analyzing dense raycasts could also be of use with other interaction techniques where accuracy is a concern.

6.3 Limitations and Future Work

We have only considered targets along the volar side of the finger of the one size within any given block. It is unclear how this would affect the performance when the density is increased around the finger, such as placing the target on the radial side of the finger or when different sizes are combined. We also have only considered the on-finger targets; off-finger targets may need additional consideration. Future work would need to further explore such different layouts and combinations. It should also be noted that our participants are young, and care should be taken when our results are further generalized to a larger population.

In contrast to touch screens, which may require indirect visualization to mitigate thumb occlusion [47, 55], HMDs offer affordances that enable enhanced visualizations. For instance, the thumb could become transparent, and a cursor could appear when the user hovers over targets. This approach could facilitate more intuitive eyes-on interactions. However, further research is needed to assess how this would impact the eyes-free affordance of thumb-to-finger interactions with HPUI [23].

7 CONCLUSION

In this paper, the interaction dynamics of Hand Proximate User Interfaces (HPUI). Our research addressed the high error rate of thumb-to-finger gestures on HPUI which shares similarities to the fat-finger problem on touch screens. By utilizing a fine-grained method of casting a multitude of rays from the thumb, we gained more granular insights into the interaction process, shedding light on how different portions of the thumb naturally interact with varying regions of the hand. Our analysis led to an enhanced interaction detection model that incorporates both spatial and temporal dimensions of thumb-to-finger interactions. We show the proposed approach improves the accuracy of denser targets in a follow-up user study. Further, our detailed analysis of thumb interaction behavior can inform future designs and implementations of thumb-to-finger-based interaction like HPUI on HMDs.

REFERENCES

[1] Ammar Ahmad, Cyrille Migniot, and Albert Dipanda. 2019. Hand Pose Estimation and Tracking in Real and Virtual Interaction:a Review. *Image and Vision Computing* 89 (2019), 35–49. https://doi.org/10.1016/j.imavis.2019.06.003

[2] Fraser Anderson and Walter F. Bischof. 2013. Learning and performance with gesture guides. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/2470654.2466143

[3] Alejandro Aponte, Arthur Caetano, Yunhao Luo, and Misha Sra. 2024. GraV: Grasp Volume Data for the Design of One-Handed XR Interfaces. In *Designing Interactive Systems Conference*. nil. https://doi.org/10.1145/3643834.3661567

[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*. https://doi.org/10.1145/3290605.3300331

[5] Anil Ufuk Batmaz and Wolfgang Stuerzlinger. 2022. Effective Throughput Analysis of Different Task Execution Strategies for Mid-Air Fitts' Tasks in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics* 28, 11 (2022), 3939–3947. https://doi.org/10.1109/tvcg.2022.3203105

[6] Jonas Blattgerste, Patrick Renner, and Thies Pfeiffer. 2018. Advantages of eye-gaze over head-gaze-based selection in virtual and augmented reality under varying field of views. In *Proceedings of the Workshop on Communication by Gaze Interaction*. nil. https://doi.org/10.1145/3206343.3206349

[7] Doug Bowman, Chadwick Wingrave, Joshua Campbell, and Vinh Ly. 2001. Using pinch gloves (tm) for both natural and abstract interaction techniques in virtual environments.

[8] Jack Brookes, Matthew Warburton, Mshari Alghadier, Mark Mon-Williams, and Faisal Mushtaq. 2019. Studying Human Behavior With Virtual Reality: the Unity Experiment Framework. *Behavior Research Methods* 52, 2 (2019), 455–463. https://doi.org/10.3758/s13428-019-01242-0

[9] Manuel Caeiro-Rodríguez, Iván Otero-González, Fernando A. Mikic-Fonte, and Martín Llamas-Nistal. 2021. A Systematic Review of Commercial Smart Gloves: Current Status and Applications. *Sensors* 21, 8 (2021), 2667. https://doi.org/10.3390/s21082667

[10] Francesco Cafaro, Leilah Lyons, and Alissa N. Antle. 2018. Framed Guessability: Improving the Discoverability of Gestures and Body Movements for Full-Body Interaction. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. nil. https://doi.org/10.1145/3173574.3174167

[11] Julia Chatain, Danielle M. Sisserman, Lea Reichardt, Violaine Fayolle, Manu Kapur, Robert W. Sumner, Fabio Zünd, and Amit H. Bermano. 2020. DigiGlo: Exploring the Palm as an Input and Display Mechanism through Digital Gloves. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*. https://doi.org/10.1145/3410404.3414260

[12] Di Laura Chen, Marcello Giordano, Hrvoje Benko, Tovi Grossman, and Stephanie Santosa. 2023. GazeRayCursor: Facilitating Virtual Reality Target Selection by Blending Gaze and Controller Raycasting. In *29th ACM Symposium on Virtual Reality Software and Technology*. https://doi.org/10.1145/3611659.3615693

[13] Taizhou Chen, Tianpei Li, Xingyu Yang, and Kening Zhu. 2022. Efring: Enabling Thumb-to-Index-Finger Microgesture Interaction through Electric Field Sensing Using Single Smart Ring. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 4 (2022), 1–31. https://doi.org/10.1145/3569478

[14] Logan D. Clark, Aakash B. Bhagat, and Sara L. Riggs. 2020. Extending Fitts' Law in Three-Dimensional Virtual Environments With Current Low-Cost Virtual Reality Technology. *International Journal of Human-Computer Studies* 139 (2020), 102413. https://doi.org/10.1016/j.ijhcs.2020.102413

[15] Jacob Cohen. 1988. *Statistical Power Analysis for the Behavioral Sciences*. Routledge. https://doi.org/10.4324/9780203771587

[16] Bastian Dewitz, Frank Steinicke, and Christian Geiger. 2019. Functional Workspace for One-Handed Tap and Swipe Microgestures. In *Mensch und Computer 2019 - Workshopband*. Gesellschaft für Informatik e.V., Bonn. https://doi.org/10.18420/muc2019-ws-440

[17] Barrett Ens, Benjamin Bach, Maxime Cordeil, Ulrich Engelke, Marcos Serrano, Wesley Willett, Arnaud Prouzeau, Christoph Anthes, Wolfgang Büschel, Cody Dunne, Tim Dwyer, Jens Grubert, Jason H. Haga, Nurit Kirshenbaum, Dylan Kobayashi, Tica Lin, Monsurat Olaosebikan, Fabian Pointecker, David Saffo, Nazmus Saquib, Dieter Schmalstieg, Danielle Albers Szafir, Matt Whitlock, and Yalong Yang. 2021. Grand Challenges in Immersive Analytics. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/3411764.3446866

[18] Barrett Ens and Pourang Irani. 2017. Spatial Analytic Interfaces: Spatial User Interfaces for in Situ Visual Analytics. *IEEE Computer Graphics and Applications* 37, 2 (2017), 66–79. https://doi.org/10.1109/mcg.2016.38

[19] Shariff AM Faleel, Michael Gammon, Kevin Fan, Da-Yuan Huang, Wei Li, and Pourang Irani. 2021. Hpui: Hand Proximate User Interfaces for One-Handed Interactions on Head Mounted Displays. *IEEE Transactions on Visualization and Computer Graphics* 27, 11 (2021), 4215–4225. https://doi.org/10.1109/tvcg.2021.3106493

[20] Shariff AM Faleel, Yishuo Liu, Roya A Cody, Bradley Rey, Linghao Du, Jiangyue Yu, Da-Yuan Huang, Pourang Irani, and Wei Li. 2023. T-Force: Exploring the Use of Typing Force for Three State Virtual Keyboards. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 723, 15 pages. https://doi.org/10.1145/3544548.3580915

[21] Shariff AM Faleel, Rajveer Sodhi, and Pourang Irani. 2024. Comparison of Unencumbered Interaction Technique for Head-Mounted Displays. *Proceedings of the ACM on Human-Computer Interaction* 8, ISS (2024), 500–516. https://doi.org/10.1145/3698146

[22] Shariff A. M. Faleel, Michael Gammon, Yumiko Sakamoto, Carlo Menon, and Pourang Irani. 2020. User gesture elicitation of common smartphone tasks for hand proximate user interfaces. In *Proceedings of the 11th Augmented Human International Conference*. https://doi.org/10.1145/3396339.3396363

[23] Shariff A M Faleel, SoonUk Kwon, David Ahlström, and Pourang Irani. 2024. Validating Eyes-free Affordance of On-Finger Hand Proximate User Interfaces in In-situ Scenarios. In *2024 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 1283–1292. https://doi.org/10.1109/ISMAR62088.2024.00145

[24] Hyunjae Gil and Ian Oakley. 2022. Thumbair: In-Air Typing for Head Mounted Displays. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 4 (2022), 1–30. https://doi.org/10.1145/3569474

[25] Álvaro González. 2009. Measurement of Areas on a Sphere Using Fibonacci and Latitude-Longitude Lattices. *Mathematical Geosciences* 42, 1 (2009), 49–64. https://doi.org/10.1007/s11004-009-9257-x

[26] Jens Grubert, Tobias Langlotz, Stefanie Zollmann, and Holger Regenbrecht. 2017. Towards Pervasive Augmented Reality: Context-Awareness in Augmented Reality. *IEEE Transactions on Visualization and Computer Graphics* 23, 6 (2017), 1706–1724. https://doi.org/10.1109/tvcg.2016.2543720

[27] Aakar Gupta, Thomas Pietrzak, Cleon Yau, Nicolas Roussel, and Ravin Balakrishnan. 2017. Summon and Select. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces*. nil. https://doi.org/10.1145/3132272.3134120

[28] Sean Gustafson, Christian Holz, and Patrick Baudisch. 2011. Imaginary Phone: Learning Imaginary Interfaces by Transferring Spatial Memory from a Familiar Device. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (Santa Barbara, California, USA) *(UIST '11)*. Association for Computing Machinery, New York, NY, USA, 283–292. https://doi.org/10.1145/2047196.2047233

[29] Ryo Hajika, Tamil Selvan Gunasekaran, Chloe Dolma Si Ying Haigh, Yun Suen Pai, Eiji Hayashi, Jaime Lien, Danielle Lottridge, and Mark Billinghurst. 2023. Radarhand: a Wrist-Worn Radar for On-Skin Touch-Based Proprioceptive Gestures. *ACM Transactions on Computer-Human Interaction* nil, nil (2023), nil.

https://doi.org/10.1145/3617365

[30] D. P. Hardin, T. J. Michaels, and E. B. Saff. 2016. A Comparison of Popular Point Configurations on S^2. *CoRR* (2016). arXiv:1607.04590 [math.NA] http://arxiv.org/abs/1607.04590v3

[31] Christian Holz and Patrick Baudisch. 2010. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. nil. https://doi.org/10.1145/1753326.1753413

[32] Christian Holz and Patrick Baudisch. 2011. Understanding touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. nil. https://doi.org/10.1145/1978942.1979308

[33] Da-Yuan Huang, Liwei Chan, Shuo Yang, Fan Wang, Rong-Hao Liang, De-Nian Yang, Yi-Ping Hung, and Bing-Yu Chen. 2016. DigitSpace: Designing Thumb-to-Fingers Touch Interfaces for One-Handed and Eyes-Free Interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. ACM, New York, NY, USA, 1526–1537. https://doi.org/10.1145/2858036.2858483

[34] Haiyan Jiang, Dongdong Weng, Zhenliang Zhang, and Feng Chen. 2019. Hifinger: One-Handed Text Entry Technique for Virtual Environments Based on Touches Between Fingers. *Sensors* 19, 14 (2019), 3063. https://doi.org/10.3390/s19143063

[35] Amy K. Karlson, Benjamin B. Bederson, and J. Contreras-Vidal. 2006. Understanding Single-Handed Mobile Device Interaction. In *Handbook of Research on User Interface Design and Evaluation for Mobile Technology*. Vol. 1. 86–101.

[36] Jong Gon Kim, Byung Geun Kim, and Seongil Lee. 2007. *Ubiquitous Hands: Context-Aware Wearable Gloves with a RF Interaction Model*. Springer Berlin Heidelberg, 546–554. https://doi.org/10.1007/978-3-540-73354-6_60

[37] Sunjun Kim and Geehyuk Lee. 2016. TapBoard 2. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/2858036.2858452

[38] Sunjun Kim, Jeongmin Son, Geehyuk Lee, Hwan Kim, and Woohun Lee. 2013. TapBoard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/2470654.2470733

[39] Marcel Krüger, Tim Gerrits, Timon Römer, Torsten Kuhlen, and Tim Weissker. 2024. Intenselect+: Enhancing Score-Based Selection in Virtual Reality. *IEEE Transactions on Visualization and Computer Graphics* nil, nil (2024), 1–10. https://doi.org/10.1109/tvcg.2024.3372077

[40] Yuki Kubo, Yuto Koguchi, Buntarou Shizuki, Shin Takahashi, and Otmar Hilliges. 2019. AudioTouch: Minimally Invasive Sensing of Micro-Gestures via Active Bio-Acoustic Sensing. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*. nil. https://doi.org/10.1145/3338286.3340147

[41] Li-Chieh Kuo, Haw-Yen Chiu, Cheung-Wen Chang, Hsiu-Yun Hsu, and Yun-Nien Sun. 2009. Functional Workspace for Precision Manipulation Between Thumb and Fingers in Normal Hands. *Journal of Electromyography and Kinesiology* 19, 5 (2009), 829–839. https://doi.org/10.1016/j.jelekin.2008.07.008

[42] Vincent Lambert, Alix Goguey, Sylvain Malacria, and Laurence Nigay. 2024. Studying the Simultaneous Visual Representation of Microgestures. In *2024 ACM International Conference on Mobile Human-Computer Interaction (MobileHCI 2024)*. Melbourne, Australia. https://doi.org/10.1145/3676523

[43] DoYoung Lee, Jiwan Kim, and Ian Oakley. 2021. FingerText: Exploring and Optimizing Performance for Wearable, Mobile and One-Handed Typing. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. nil. https://doi.org/10.1145/3411764.3445106

[44] Lik-Hang Lee and Pan Hui. 2018. Interaction Methods for Smart Glasses: a Survey. *IEEE Access* 6 (2018), 28712–28732. https://doi.org/10.1109/access.2018.2831081

[45] Rui Li, Zhenyu Liu, and Jianrong Tan. 2019. A Survey on 3d Hand Pose Estimation: Cameras, Methods, and Datasets. *Pattern Recognition* 93 (2019), 251–272. https://doi.org/10.1016/j.patcog.2019.04.026

[46] Yiqin Lu, Chun Yu, and Yuanchun Shi. 2020. 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)*. nil. https://doi.org/10.1109/vr46266.2020.00021

[47] Ali Neshati, Aaron Salo, Shariff Am Faleel, Ziming Li, Hai-Ning Liang, Celine Latulipe, and Pourang Irani. 2022. EdgeSelect: Smartwatch Data Interaction with Minimal Screen Occlusion. In *INTERNATIONAL CONFERENCE ON MULTIMODAL INTERACTION*. https://doi.org/10.1145/3536221.3556586

[48] Francisco Perella-Holfeld, Shariff AM Faleel, and Pourang Irani. 2023. Evaluating design guidelines for hand proximate user interfaces. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference*. https://doi.org/10.1145/3563657.3596117

[49] Yuan Yuan Qian and Robert J. Teather. 2017. The eyes don't have it: An Empirical Comparison of Head-Based and Eye-Based Selection in Virtual Reality. In *Proceedings of the 5th Symposium on Spatial User Interaction*. https://doi.org/10.1145/3131277.3132182

[50] Katharina Reiter, Ken Pfeuffer, Augusto Esteves, Tim Mittermeier, and Florian Alt. 2022. Look & Turn: One-handed and Expressive Menu Interaction by Gaze and Arm Turns in VR. In *2022 Symposium on Eye Tracking Research and Applications*. nil. https://doi.org/10.1145/3517031.3529233

[51] Rongkai Shi, Jialin Zhang, Yong Yue, Lingyun Yu, and Hai-Ning Liang. 2023. Exploration of Bare-Hand Mid-Air Pointing Selection Techniques for Dense Virtual Reality Environments. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/3544549.3585615

[52] Mohamed Soliman, Franziska Mueller, Lena Hegemann, Joan Sol Roo, Christian Theobalt, and Jürgen Steimle. 2018. FingerInput: Capturing Expressive Single-Hand Thumb-to-Finger Microgestures. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces* (Tokyo, Japan) *(ISS '18)*. ACM, New York, NY, USA, 177–187. https://doi.org/10.1145/3279778.3279799

[53] Sakari Tamminen, Antti Oulasvirta, Kalle Toiskallio, and Anu Kankainen. 2004. Understanding Mobile Contexts. *Personal and Ubiquitous Computing* 8, 2 (2004), 135–143. https://doi.org/10.1007/s00779-004-0263-1

[54] Hsin-Ruey Tsai, Te-Yen Wu, Da-Yuan Huang, Min-Chieh Hsiu, Jui-Chun Hsiao, Yi-Ping Hung, Mike Y. Chen, and Bing-Yu Chen. 2017. SegTouch: Enhancing Touch Input While Providing Touch Gestures on Screens Using Thumb-To-Index-Finger Gestures. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI EA '17)*. ACM, New York, NY, USA, 2164–2171. https://doi.org/10.1145/3027063.3053109

[55] Daniel Vogel and Patrick Baudisch. 2007. Shift: A Technique for Operating Pen-Based Interfaces Using Touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. nil. https://doi.org/10.1145/1240624.1240727

[56] Anandghan Waghmare, Youssef Ben Taleb, Ishan Chatterjee, Arjun Narendra, and Shwetak Patel. 2023. Z-Ring: Single-Point Bio-Impedance Sensing for Gesture, Touch, Object and User Recognition. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–18. https://doi.org/10.1145/3544548.3581422

[57] Dennis Wolf, Jan Gugenheimer, Marco Combosch, and Enrico Rukzio. 2020. Understanding the Heisenberg Effect of Spatial Interaction: A Selection Induced Error for Spatially Tracked Input Devices. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/3313831.3376876

[58] Zheer Xu, Pui Chung Wong, Jun Gong, Te-Yen Wu, Aditya Shekhar Nittala, Xiaojun Bi, Jürgen Steimle, Hongbo Fu, Kening Zhu, and Xing-Dong Yang. 2019. TipText: Eyes-Free Text Entry on a Fingertip Keyboard. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. ACM, New York, NY, USA, 883–899. https://doi.org/10.1145/3332165.3347865

[59] Difeng Yu, Hai-Ning Liang, Xueshi Lu, Kaixuan Fan, and Barrett Ens. 2019. Modeling Endpoint Distribution of Pointing Selection Tasks in Virtual Reality Environments. *ACM Transactions on Graphics* 38, 6 (2019), 1–13. https://doi.org/10.1145/3355089.3356544

[60] Difeng Yu, Qiushi Zhou, Tilman Dingler, Eduardo Velloso, and Jorge Goncalves. 2022. Blending On-Body and Mid-Air Interaction in Virtual Reality. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. https://doi.org/10.1109/ismar55827.2022.00081

[61] Tianhong Catherine Yu, Guilin Hu, Ruidong Zhang, Hyunchul Lim, Saif Mahmud, Chi-Jung Lee, Ke Li, Devansh Agarwal, Shuyang Nie, Jinseok Oh, François Guimbretière, and Cheng Zhang. 2024. Ring-A-pose: a Ring for Continuous Hand Pose Tracking. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 8, 4 (2024), 1–30. https://doi.org/10.1145/3699741

[62] Cheng Zhang, Anandghan Waghmare, Pranav Kundra, Yiming Pu, Scott Gilliland, Thomas Ploetz, Thad E. Starner, Omer T. Inan, and Gregory D. Abowd. 2017. FingerSound: Recognizing unistroke thumb gesture using a ring. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 1–19. https://doi.org/10.1145/3130985

[63] Ce Zheng, Wenhan Wu, Chen Chen, Taojiannan Yang, Sijie Zhu, Ju Shen, Nasser Kehtarnavaz, and Mubarak Shah. 2023. Deep Learning-Based Human Pose Estimation: a Survey. *Comput. Surveys* 56, 1 (2023), 1–37. https://doi.org/10.1145/3603618

A APPENDIX

We provide additional analysis from the data collected in Section 3. Figure 18 shows the number of rays used to select the index distal (see Figure 6) target in the sparse layout for each participant. Note that each target in the sparse condition was selected 20 times. Figure 19 shows the same for the proximal phalanx of the index finger in the sparse layout condition (see Figure 6).

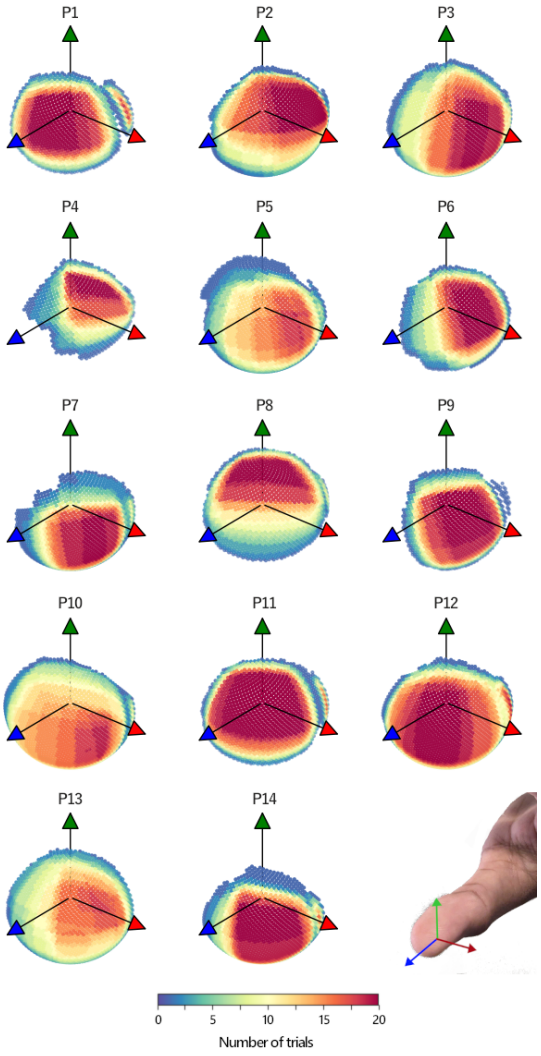


Figure 18: Each plot shows the rays used when selecting the Index Distal target in the sparse layout (see Figure 6) for each participant. For each participant, the plot shows the number of trials during which a ray was in contact with the intended target. The rays are from the frame where the highest number of rays were in contact with the intended target during a given tap gesture.

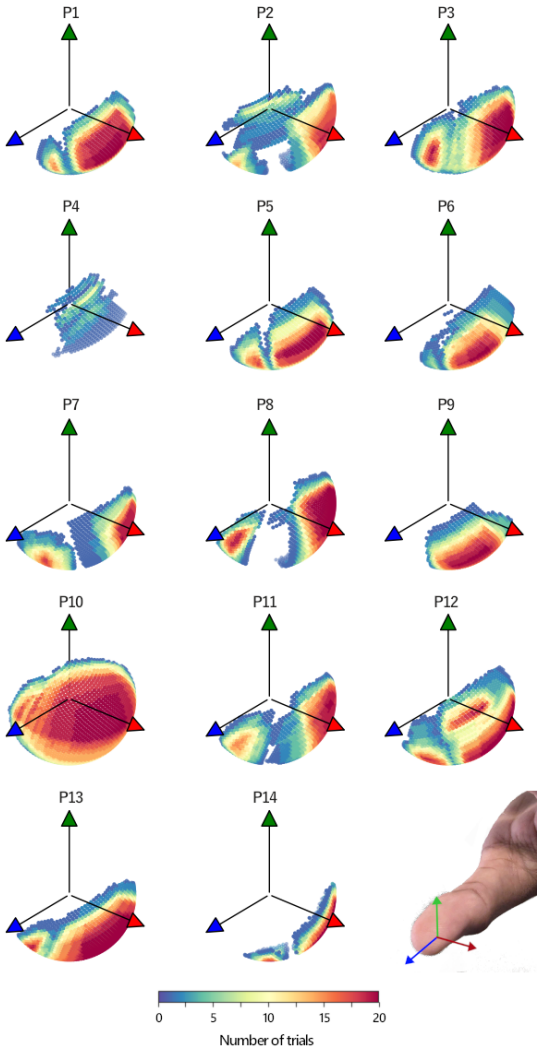


Figure 19: Each plot shows the rays used when selecting the Index Proximal target in the sparse layout (see Figure 6) for each participant. For each participant, the plot shows the number of trials during which a ray was in contact with the intended target. The rays are from the frame where the highest number of rays were in contact with the intended target during a given tap gesture.

Given the observation that not all rays are used when selecting a given target, we can reduce the number of the ray cast based on which target is to get selected. The Figure 20 shows the "cones" of raycasts when interacting with a target on a given phalanx. The centroid black arrow is the mean direction of all rays used to select any of the targets by any participant. The threshold of the cone was computed by getting the 90% percentile ray direction. This further validates our observation in Section 3.5.2 that the region of the thumb that gets used to select a target differs based on which phalanx the thumb is interacting with.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009

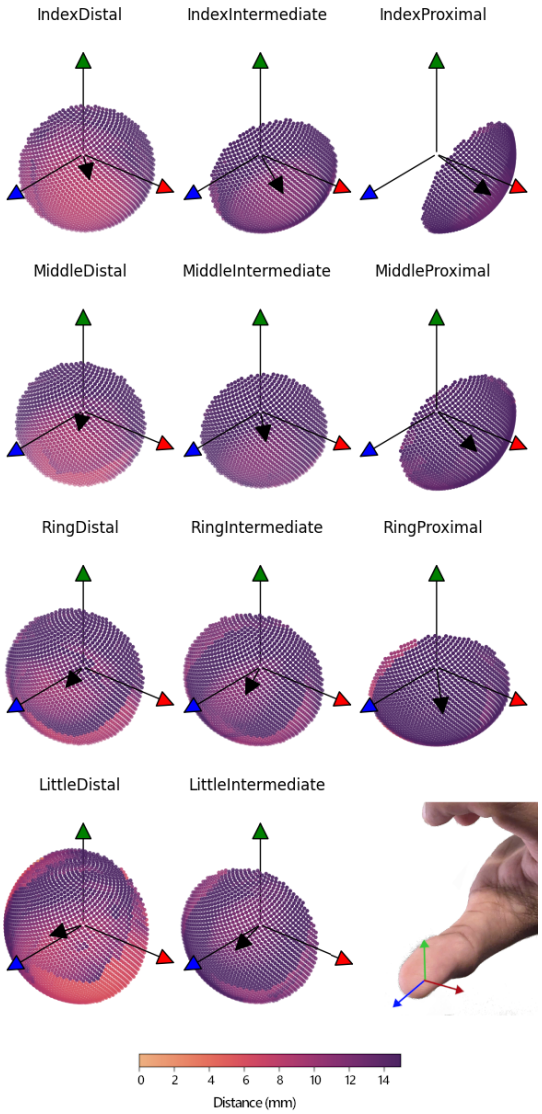


Figure 20: The "cone" of rays extracted from the all the rays interacted with a given target placed in the respective phalanx. The black arrows show the centroid of the "cone".