Selecting Real-World Objects via User-Perspective Phone Occlusion

Yue Qin
Tsinghua University
Beijing, China
qiny19@mails.tsinghua.edu.cn

Chun Yu
Tsinghua University
Beijing, China
chunyu@tsinghua.edu.cn

Wentao Yao
Tsinghua University
Beijing, China
yaowt19@mails.tsinghua.edu.cn

Jiachen Yao
Tsinghua University
Beijing, China
yaojc20@mails.tsinghua.edu.cn

Chen Liang
Tsinghua University
Beijing, China
liangchenc@163.com

Yueting Weng
Tsinghua University
Beijing, China
wengyt19@mails.tsinghua.edu.cn

Yukang Yan
Tsinghua University
Beijing, China
yanyukanglwy@gmail.com

Yuanchun Shi
Tsinghua University
Beijing, China
shiyc@tsinghua.edu.cn

ABSTRACT

Perceiving the region of interest (ROI) and target object by smartphones from the user's first-person perspective can enable diverse spatial interactions. In this paper, we propose a novel ROI input method and a target selecting method for smartphones by utilizing the user-perspective phone occlusion. This concept of turning the phone into real-world physical cursor benefits from the proprioception, gets rid of the constraint of camera preview, and allows users to rapidly and accurately select the target object. Meanwhile, our method can provide a resizable and rotatable rectangular ROI to disambiguate dense targets. We implemented the prototype system by positioning the user's iris with the front camera and estimating the rectangular area blocked by the phone with the rear camera simultaneously, followed by a target prediction algorithm with the distance-weighted Jaccard index. We analyzed the behavioral models of using our method and evaluated our prototype system's pointing accuracy and usability. Results showed that our method is well-accepted by the users for its convenience, accuracy, and efficiency.

CCS CONCEPTS

• **Human-centered computing** → **Pointing**; *Human computer interaction (HCI)*; Interaction techniques.

KEYWORDS

object selection, smartphone interaction

CHI '23, April 23–28, 2023, Hamburg, Germany

ACM ISBN 978-1-4503-9421-5/23/04.
https://doi.org/10.1145/3544548.3580696

ACM Reference Format:
Yue Qin, Chun Yu, Wentao Yao, Jiachen Yao, Chen Liang, Yueting Weng, Yukang Yan, and Yuanchun Shi. 2023. Selecting Real-World Objects via User-Perspective Phone Occlusion. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), April 23–28, 2023, Hamburg, Germany*. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3544548.3580696

1 INTRODUCTION

Mobile computing allows us to quickly and easily connect and interact with a large number of nearby ubiquitously distributed appliances or get information from nearby objects. Using the camera to perceive the physical world is an intuitive way to enable the phone to interact with in-sight objects. For example, after confirming the interaction target via the rear camera, the smartphone can directly trigger the APP function bound to the target [15, 17] (e.g., scanning the QR code, issuing the user-defined command, or displaying a control interface), or perform multi-modal interaction combined with voice and gestures [43]. One of the key issues is how to make the smartphone quickly and accurately identify the target the user sees.

Traditional methods offer two types of solutions. The first is to actively turn on the camera and render the camera preview on the screen. The user confirms the on-screen target by tapping it or pointing an on-screen selector (such as a crosshair) at the target. These methods usually require multiple steps, such as opening the camera preview, waiting for the screen to render the camera preview, aligning the camera preview to the target, confirming the target, and finally speaking voice commands or interacting with gestures. This multi-step approach can be heavy if a user wants to perform a quick one-shot interaction (e.g., asking "how much is that"). Some past works showed that reducing the explicit wake-up steps or the visual dependence of the screen can significantly improve the interaction efficiency and user experience [50, 57, 66, 67]. Extending this idea to camera-based target selection, the second type of method allows the user to directly confirm the target through the direction the mobile phone camera is pointing or the direction the

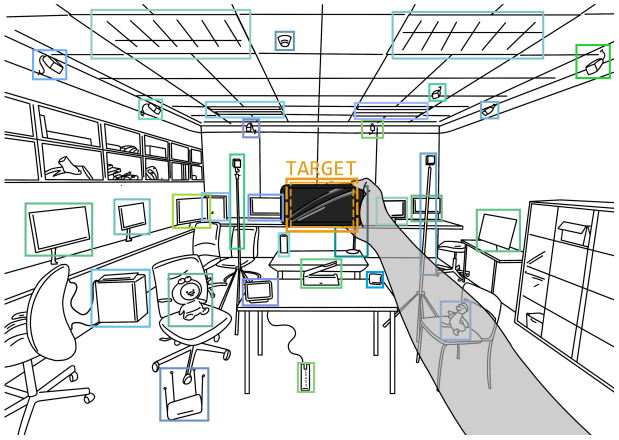


Figure 1: Phone-occlusion-based object selection technique. A user raises the phone and approximately blocks the target from the first-person perspective.

face is facing while the screen is off [31, 43]. This method greatly simplifies the interaction process and allows users to directly input voice commands or gestures after raising the phone. However, such methods face the problem of inaccurate pointing due to the lack of visual feedback, which exacerbates the tensions and insecurities of the users [20, 48]. Our work aims to improve the efficiency and accuracy of spatial interaction with smartphones without opening the camera preview and reduce user insecurity due to the lack of visual feedback.

In this paper, we propose to use a novel selection tool to simultaneously improve the efficiency, accuracy, and user experience of interacting with in-sight objects using smartphones, i.e., a resizable and rotatable rectangular region of interest (ROI) provided by the user-perspective phone occlusion. When the users want to interact with the objects in sight, they raise the phone, block the target from their perspective, and trigger the gesture commands or speak directly to the voice assistant. Our approach benefits from three parts. First, this posture makes it easy for the front and rear cameras to capture the face and target to sense the environment. Second, this approach gets rid of the constraint of camera preview by utilizing the visual feedback provided by the phone case. This allows users to easily be confident that they have selected the target accurately while omitting the interactive steps related to the camera preview. Third, users can freely rotate or move the phone closer to or further away from the face to get a resizable and rotatable rectangular ROI. Compared to ray-based methods (e.g., pointing or gazing), using the resizable and rotatable rectangular ROI as the area cursor can easily select sparse targets. At the same time, utilizing the similarity of the ROI and the geometric features of the target to disambiguate dense or overlapping scenarios is potential.

In this paper, we mainly study the following three questions:

RQ1: How do users mentally map the occluded area (ROI) to a specific target?

RQ2: Based on the existing state-of-the-art algorithm, what is the pointing accuracy of our method?

RQ3: What is the difference in user experience between our method and traditional methods?

In this work, we first developed an algorithm to calculate the rectangular area occluded by the phone through the images of the front and rear cameras. Then by collecting and analyzing user occlusion behaviors, we designed a target prediction algorithm to map the occlusion rectangle to the target object with the distance-weighted Jaccard index. Then we conducted two user studies to evaluate the accuracy of the occlusion area estimation algorithm and the user experience when using our prototype system. The results showed that our prototype system could achieve an average pointing error of 1.28°±0.96°, and users generally agreed that the occlusion-based target selection technique is convenient, accurate, and efficient.

2 RELATED WORK

2.1 The 3D Target Selection Techniques

Many different 3D target selection techniques are designed for different application scenarios and devices. The survey by Argelaguet et al. divides them into several categories according to different characteristics [5].

According to the selection tool, the target selection technique can be classified into the virtual hand, ray-casting, and area/volume cursor. Using virtual hands to touch the objects directly is proved efficient when the user is close to the target [22, 52, 64]. Using ray-casting combined with visual feedback can effectively select small, dense, and far objects [29, 37]. The area cursor can quickly select sparse objects [26, 37], but additional disambiguation steps are required if multiple objects are in the selection area [18].

Given the different starting points, the 3D target selection technique could be divided into several categories, including: 1) Body-centered ray-casting [45–47], such as finger-rooted ray cast [11, 32], head-gaze ray cast [43, 65, 70], eye-gaze ray cast [69], eye-finger ray cast [20, 40], or a combination of above [34, 56]. 2) Device-centered ray-casting, which leverages the orientation of device (e.g., controller [41], smartwatch [3], and mobile phone[2, 59])).

Our method is similar to eye-finger ray-casting but at the same time has the feature of an area cursor to select sparse targets quickly. In addition, we introduce a disambiguation mechanism using ROI geometric similarity (distance-weighted Jaccard index) to further support dense and overlapping objects.

2.2 Target Selection on Smartphones

Speed and accuracy are critical indicators when using smartphones to perform one-shot interactions with in-sight objects. Most of the above methods cannot be used for smartphones due to smartphones' lack of sensing capabilities or feedback mechanisms. The head-rooted or device-centered ray-casting is mainly considered in previous works.

Device-centered approaches often require rendering the camera preview on the phone screen. To select the target, the user taps the target on the screen [12, 62] or finely adjusts the orientation of the phone to align the on-screen selection tool at the target (e.g., the crosshair rendered in the center of the screen) [54, 55]. This approach has been applied to many augmented reality (AR) scenarios [28, 68]. However, such methods relying heavily on visual feedback

can compromise the efficiency and smooth user experience due to the multiple interaction steps required before issuing voice/gesture commands.

For the head-rooted approach, previous work suggested using head orientation or eye gaze to select the target for voice input with smartphones [43]. If the camera preview and the gaze-ray are not rendered on screen, such methods will face the problem of pointing inaccurately. Mayer et al. reported that the state-of-the-art algorithms could only achieve around ±10° angular error for the head-gaze and around ±15°angular error for the eye-gaze to select distant targets using the smartphone [43, 44]). This inaccuracy comes from two reasons. For head gaze, people feel it difficult to perceive the actual orientation of the head and feel strained [36]. For the eye gaze, a slight iris shift in the image may bring about a considerable gaze direction change, which is computationally unfriendly [33].

Our approach benefits from the use case of the handheld smartphone. Using the front and rear cameras to perceive the occlusion area, we can effectively avoid the above two problems and provide higher accuracy (around ±1.28° for ray-casting). On the one hand, our method does not require the user to control the orientation of the head and the phone finely. On the other hand, we only need to estimate the 2D coordinates of the iris relative to the center of the front image to get the eye-phone virtual ray rather than estimate the slight offsets of the iris relative to the eye for sensing eye-gaze, which will be discussed in the next section.

2.3 User-Perspective Interaction

The most related works to our work are user-perspective 3D object selection techniques. These methods use the field of view of the user's eyes as a 2D interaction plane. The previous works can mainly divide into two categories, image plane metaphor (similar to eye-finger ray-casting) and magic lenses paradigm (a class of see-through interfaces or transparent area cursor). Image plane techniques require the users to align the target with a hand-held aperture [20] or with their fingers [7, 40, 49]. Magic Lenses work by overlaying a transparent tool glass onto the target to reveal hidden information, enhance data of interest, or suppress distracting information [9, 39, 42, 61].

User-perspective techniques have proven to be more natural and efficient than the device-centric approaches because "we do not have to live with the phone's eyes" [8, 60]. However, the double-vision problem is an important issue that restricts the use of user-perspective techniques in the real world; that is, it almost impossible to "align the target with the user's finger" in the real world. For example, when the user's gaze is focused on the distant object, the closer finger will split into two ghosts. Conversely, if the user's gaze is focused on the closer finger, the distant object will be split into two ghosts. This means that the users cannot know which selection tool to use unless they closes one eye or randomly chooses one of the two ghosts according to the dominant eye effect. Not only the ray-based user's perspective techniques but also the transparent magic lenses face such problem.

Our approach is different from all of the above. We recommend using an 'opaque' occlusion rectangle as the selection tool to solve the double-vision problem in the real world. When the user is looking at the distant target, although the selection tool will split into two ghosts, only one occluded area is invisible to the user, i.e., the intersection area of the two rectangle ghosts. This occlusion area is easily understood and recognized by the user. Additionally, we study how to utilize the resizable and rotatable rectangular area cursor to disambiguate dense or overlapping scenarios, which is suitable for use with smartphones.

3 ALGORITHM AND IMPLEMENTATION

In this section, we introduce the principles of occlusion-based object selection via smartphones. We designed a three-stage pipeline to implement the occlusion-based object selection system on the mobile phone. The three components of the pipeline are occlusion rectangle estimation, object detection, and target prediction. We opened the front and rear cameras to take pictures simultaneously.

3.1 Occlusion Rectangle Estimation

The occlusion rectangle estimation algorithm aims to estimate the rectangular area which is not visible to the user in the rear camera image of the phone. We first use the MediaPipe Iris [1, 23] to locate the 3D positions of the user's irises, which uses the RGB image from the front camera of the phone with a depth error of less than 10% [24]. And then, according to the known and fixed geometric relationship between the front and rear cameras, we can obtain the occlusion area under the user's perspective in the rear camera image. Figure 2 and Equation 1 show the calculation principle of the occlusion rectangle estimation.

$$T_x = (D_x - D_c) + (D_x - I_x) * \frac{T_z}{I_z} \tag{1}$$

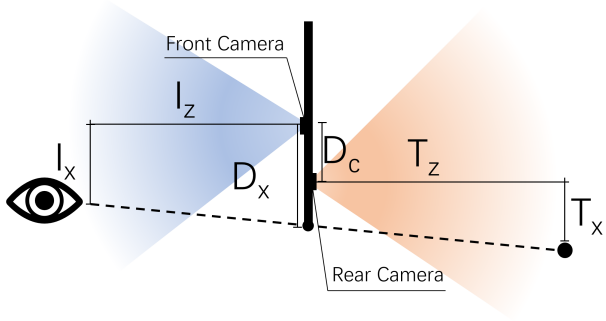
By locating the 2D pixel coordinates of the iris from the front camera image, $\frac{I_x}{I_z}$ can be obtained. The eye-phone distance I_z can be estimated by MediaPipe Iris [23]. D_x is the fixed distance between the corner of the phone and the front camera. D_c is the fixed distance between the front and rear cameras. The D_x and D_c can be obtained when the phone is produced. We use the rear dual camera of the phone to estimate the target-phone distance T_z. The above parameters can also be estimated with the smartphones' front/rear true depth cameras (e.g., structured-light and LiDAR for depth sensing).

Analyzing the theoretical estimation error of our method is useful for understanding its theoretical accuracy. Consider Equation 1, we can get the error formula of the estimator as Equation 2, where $\frac{I_x}{I_z}$ is the 2D x-coordinate of the iris in the front-camera image (the y-coordinate is similar), and $\frac{T_x}{T_z}$ is the x-coordinate of a corner of the occlusion rectangle in the rear-camera image.

$$\mathrm{d}\frac{T_x}{T_z} = -\mathrm{d}\frac{I_x}{I_z} + (\frac{1}{T_z} + \frac{1}{I_z})\mathrm{d}D_x + \frac{(D_c - D_x)}{T_z^2}\mathrm{d}T_z - \frac{D_x}{I_z^2}\mathrm{d}I_z \tag{2}$$

Considering the typical values of practically applicable scenarios (e.g., D_x, I_z, T_z are around 5cm, 30cm, 400cm), we can find that the effect of $\mathrm{d}T_z$ can be approximately ignored due to its small effect on estimation error. In practice, estimating the depth of the iris (i.e., I_z, around 10% estimation error [24]) is inaccurate compared to estimating the coordinates of the iris in the front camera image (i.e.,

(a) Front Image (b) Rear Image



(c) Geometric Relationship

Figure 2: The geometric schematic diagram of the projection point from the iris to the rear-camera image. The orange, blue, and red rectangles represent the right eye, the left eye, and the user's overall invisible area.

I_x/I_z, around 0.3° estimation error). According to the above analysis, it can be roughly concluded that the theoretical estimation accuracy of our approach is around 1°. In particular, when $D_x = D_c = 0$, the estimation error is only affected by the 2D coordinate of the iris in the front image (i.e., $\frac{I_x}{I_z}$), but not by the depth of I_z and T_z. In this case, the error can be further reduced to around 0.3°. The actual pointing accuracy will be measured in Study 2.

3.2 Object Detection

After obtaining the occlusion rectangle in the rear camera image, we performed the object detection algorithm to locate the bounding boxes of all interactable objects. For the object detection or recognition tasks, neural networks and deep learning have shown strong performance in recent years [19, 21, 38, 51, 53], and there have been several models which can run on mobile phones in real-time [14, 30]. As a proof of concept, we used the YOLOv4 framework [10] as the object detection backend in our prototype system.

3.3 Target Prediction

The occlusion rectangle denotes a region of interest (ROI), but it does not immediately provide a well-defined object of interest. In our target prediction algorithm, we use Bayes' theorem (Equation 3) to estimate the probability of each candidate object.

$$P(T_k|O) \propto P(O|T_k) * P(T_k) \quad (3)$$

In Equation 3, O represents the occlusion rectangle, T_k is the k-th target in the rear image. $P(T_k)$ is the prior probability of selecting the k-th target. We assume that $P(T_k)$ is equal for all objects. From the equation, we can find that the target prediction algorithm relies on the modeling of the user's behavior model $P(O|T_k)$ which will be discussed in Study 1, that is, the probability distribution of the occlusion rectangle when the user wants to select a certain object.

3.4 Implementation

We implemented our system on the iPhone 12 Pro, which had a width of 7.1 cm, a height of 14.6 cm, and a weight of 187g. We used its rear wide-angle dual camera to capture 70° field of view (FOV) RGB-D images and its front camera to capture 56° FOV RGB images. At the same time, according to the gravity direction sensed by the mobile phone's intrinsic measurement unit (IMU), we rotated and normalized the camera images to facilitate iris tracking and object recognition.

As a proof of concept, we built an application to take photos on the mobile phone and deployed the iris tracking and object recognition algorithm on a server in the local area network. The server has an Intel(R) Xeon(R) E5-2640 v4 @ 2.40GHz CPU and a TITAN Xp GPU for calculations. The neural network models used for iris tracking and object recognition consume 259 MB of Memory. The average time for image processing is around 731ms, including 498ms for network transmission, 138ms for iris tracking on one core of CPU, 28ms for object detection on a single GPU, and 67ms for other calculations on CPU.

4 STUDY 1: UNDERSTANDING USER BEHAVIOR

We first conducted a user study to investigate users' target selection behavior with phone occlusion, collecting data to analyze and model the behavioral term in Equation 3. Since it is easy to predict which objects the user wants to select for scenes with sparse targets, we mainly focus on the user's behavior in dense and overlapping scenarios. In this study, we aim to collect: 1) images from the user's perspective (containing the opaque phone), 2) images from the user's perspective when the phone is transparent (to get the area that the user cannot see due to the opaque phone), 3) images from the phone's rear camera and 4) the 3D coordinates and orientations of two eyes, the phone and all objects in the scene. Since we cannot acquire some of the data above simultaneously in a real-world setting (e.g., the phone area and the occluded area from the user's perspective), we conducted this study in a simulated VR environment refer to the previous work [16, 35]. We built a VR application and virtual scene to collect the above data. The data is analyzed offline, so there are no real-time target selection algorithms running in the VR application.

4.1 Participants

We recruited 11 participants (7 male and 4 female) from our institution. The average age of participants was 22.7 years. All of them were right-handed. The whole study took around 20 minutes for each participant. Each participant received $10 for compensation.

4.2 Apparatus and Platform

We built a virtual room-scale scenario containing 49 common objects in VR (shown in Figure 3), and used the HTC Vive VR headset to render the environment. We fixed a VIVE Tracker on the phone case to simulate a real phone which is similar to the work of Bai et al. [6]. The simulated phone (including the phone case and the tracker) weighs 130g, whose gripping sense is similar to a real phone, and the size is the same as iPhone 12 Pro. The simulated phone is similar in weight to a real phone, but the weight distribution is slightly different. The size of virtual objects in VR is the same as real-world ones. We also used a Vive controller to control the experiment process.

(a) VR Scene

(b) Experiment Environment

(c) VR Phone

Figure 3: The VR indoor scene from the participant's first-person perspective, experiment environment, and simulated phone. The targets are marked with different colors to facilitate readers to distinguish.

4.3 Task

The tasks are composed of three sessions. In each session, the user stands in a specific position to select 49 objects in the room, as shown in Figure 3. Each object is required to be selected one time in one session. The order of selection is randomly shuffled. In total, there are 147 (3×49) tasks for each participant.

4.4 Procedure

We first described the concept of occlusion-based target selection method. Before the experiment, participants got themselves familiarized with the operations. They then followed instructions and performed operations step by step. We told the participants that they could complete tasks naturally according to their own understanding. Participants were also told that they should try to make it possible for others to predict their target based on their phone's location. In other words, the participants were not told the specific rules and standards of the occlusion interaction, but "blocked" the target naturally, quickly, and accurately according to their understanding. At the beginning of each task, a red arrow widget is displayed at the center of the screen indicating the position of the target. Meanwhile, the contour of the target is constantly flickering. When the participant confirms the target, he should press the button on the Vive controller to start collecting data, and the target will no longer flash. When the participant raises the phone and blocks the target, he needs to press the button again to end the recording. A two-minute break was placed after each session. During the experiment, The experimenter frequently asked participants *"Why do you hold your phone in that specific position"* to obtain the users' thoughts and feedback.

4.5 Analysis of the results

We collected the data of 1617 selections from 11 participants. On average, each selection took 1.39 seconds for each participant. All participants reported that the gripping feelings of the simulated phone in VR had no difference from that of a real-world phone.

Because the positions of the two iris are different, there will be two different occlusion rectangles in the rear camera image corresponding to the left eye and the right eye. The first issue we care about is how users will deal with this double-vision problem. Users reported that they were more inclined to stare at distant objects rather than focus on the nearby phone screen and used the "invisible area" as a psychological selection box. We have two assumptions about the "invisible area." The first one is that the user only adopts the image seen by the dominant eye during the cognitive process. Under this assumption, the invisible area corresponds to the occlusion rectangle of a specific eye. The second assumption is that the "invisible area" is the area the user with either eye cannot see. Under this assumption, the invisible area corresponds to the intersection area of the two occlusion rectangles formed by the left and right eyes. From Figure 4a, we found that the data are clustered into one cluster, and the data for any user under the dominant eye assumption is off-center. The above evidence leads us to believe that ocular dominance plays a small role in our method and could be ignored. Therefore, in the following, we define *occlusion rectangle* as the intersection of two rectangular areas in the rear camera image.

Our second focus is where the users will place the occlusion rectangle (i.e., the phone) in their field of view. From Figure 4b, we observed that users generally held the phone vertically or horizontally, and rarely tilted it. Through interviews with users, we found that users tended to spend less effort to achieve their purposes. Sometimes the users did not raise the phone to fully overlap the target but held it lower than the target a little bit to save physical effort. Therefore, the coordinates of the center point in Figure 4a will be slightly lower.

On the whole, users had similar thoughts and behaviors, from which we concluded our observations with three points: 1) Users believed that the most accurate position is to align the phone's

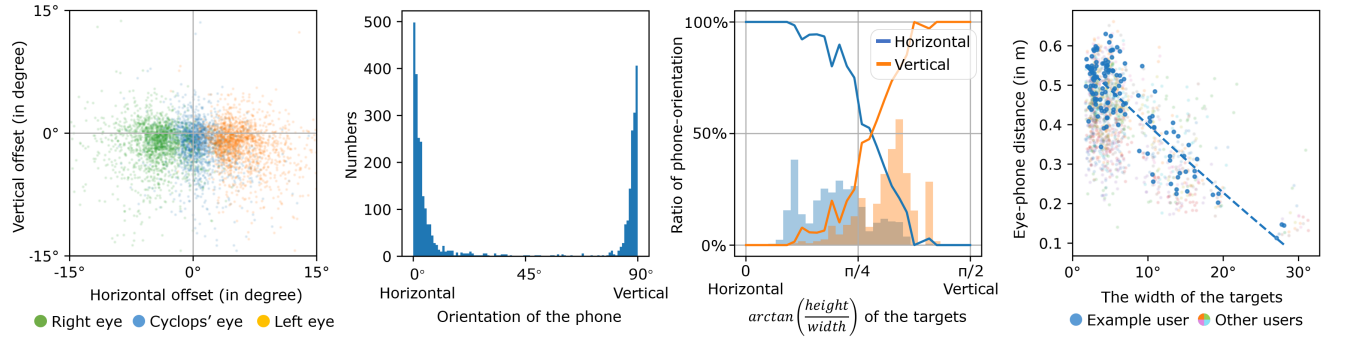


Figure 4: Characteristics of user behavior. (a) shows the offset between the center of the occlusion rectangle and the center of the target under the three assumptions separately, i.e., left/right eye only and the intersection of two occlusion rectangles generated by the left and right eyes respectively (Cyclops' eye). (b) shows that users mostly use horizontal or vertical phone orientation. (c) shows how many ratios of users tend to occlude the target horizontally or vertically as the aspect ratios of target change (d) shows the relationship between the user's eye-phone distance and the target size. The data of user-1 in (d) is highlighted to reflect intra-user and cross-user trends.

center with the target's center (as shown in 4a); 2) Users tended to rotate the phone to match the object's main axis (as shown in 4c); It means holding the phone horizontally for flat and wide objects, and holding the phone vertically for thin and tall objects. 3) Users deemed that putting the phone closer/further to their eyes to select larger/smaller targets made sense (as shown in 4d).

Considering the above factors, we chose distance-weighted Jaccard index to establish our target prediction algorithm (shown in Equation 4). It can reflect on the consistency of two rectangles, including 1) the center-to-center offset between the phone and target, 2) the orientation of the phone, and 3) the size of the occluded rectangle. While it is feasible to represent the target as an irregularly shaped mask and model its probability distribution, as a proof-of-concept, we only approximate the target as a rectangular bounding box. Firstly, we considered the rectangular bounding box of the object T_k and the occlusion rectangle of the phone O as two-dimensional uniform distribution. We then keep its mean vector μ and covariance matrix Σ and transform them into Gaussian function f_k and g with the maximum value of 1. We use the weighted Jaccard index $J_{\mathcal{W}}$ to measure the similarity between the occlusion rectangle and a single target, which is widely used to measure the similarity between sets and geometric shapes. When the occlusion rectangle and the target completely coincide, $J_{\mathcal{W}}$ reaches its maximum value of 1.

$$
\begin{aligned}
f(\mathbf{x}) &= \exp(-(\mathbf{x}-\mu_o)^T\Sigma_o^{-1}(\mathbf{x}-\mu_o)) \\
g_k(\mathbf{x}) &= \exp(-(\mathbf{x}-\mu_k)^T\Sigma_k^{-1}(\mathbf{x}-\mu_k)) \\
J_{\mathcal{W}}(f,g_k) &= \frac{\int \min(f,g_k)d\mu}{\int \max(f,g_k)d\mu}
\end{aligned}
\tag{4}
$$

A trick we found to effectively improve the accuracy of object prediction with distance-weighted Jaccard index is to inflate small objects. Due to the limited length of users' arm, it's hard to get a rectangle small enough to exactly cover a small and distant target. Therefore, we empirically zoomed those small-size bounding boxes to $\frac{w*h}{f_x*f_y} = 1*10^{-2}$ and kept their aspect ratio, where w and h is the width and height of the bounding box, f_x and f_y is the focal length of the camera intrinsic matrix. In our target prediction algorithm, we use $J_{\mathcal{W}}(f,g_k)$ to approximate $P(O|T_k)$. We finally compare the shape similarity between the occlusion rectangle and all targets and choose one with the highest probability.

4.6 Evaluation of Behavior Model

We evaluated the accuracy of our target prediction algorithm with the other three existing white-box baselines on the data set. All setups were the same except the similarity metrics. The four methods are as follows:

Center-to-Area Distance (C2A) . This implementation is similar to the bubble cursor[25], which uses the closest distance from the center point of the occlusion area to the target as the metric. If the center of the occlusion rectangle is inside multiple objects, we choose the object with the smallest area.

Center-to-Center Distance (C2C) . This implementation uses the distance from the center point of the occlusion area to the center point of the target as the metric.

Intersection over Union (IoU) . This implementation uses the IoU ratio of two rectangles as the metric.

Our method. This implementation uses the distance-weighted Jaccard index $J_{\mathcal{W}}$ as the metric (described in Equation 4).

Table 1 shows that our method (accuracy: 96.6%) outperformed the baselines. The center-to-center distance also shows good performance.

Table 1: The accuracy of the four similarity metrics on the evaluation data set.

Method	C2A	C2C	IoU	Ours
Accuracy	91.5%	92.6%	84.0%	96.6%

5 STUDY 2: EVALUATION OF POINTING ERROR

This experiment is used to study the user's ability to control the occlusion area and evaluate our prototype system's average pointing error of the occluded area estimation. We collected images captured by smart phones and analyze it offline. The overall selection accuracy of our prototype system will be evaluated in Study 3.

5.1 Participants

We recruited 12 participants (10 male and 2 female) from our institution. The average age of participants was 21.4 years, with an average height of 173.2±6.1 cm, and an average arm length of 53.2±4.8cm. All of them were right-handed and familiar with smartphones. The whole study took around 20 minutes and each participant received $10 for compensation.

5.2 Apparatus

We conducted this experiment in a real room instead of in VR. We used iPhone 12 Pro as the experimental device, which always remained a black screen. We hung a 15cm × 15cm crosshair target on the wall, which was 150cm above the ground. This study was conducted in a bright and clean environment. We did not use other equipment except the smartphone and the crosshair target.

5.3 Design and Procedure

The experimenter first gave a brief introduction to participants, asking for their demographic information and answering their questions. During the experiment, participants were asked to stand between 1m and 4m from the target and were required to hold the phone and raise it naturally with their dominant hand. When the corner of the phone (i.e., the intersection of two edges) was aimed at the center of the crosshair from their perspective, they had to tap on the touchscreen to capture two images by the front and rear cameras simultaneously. We did not provide participants feedback, so they could not learn during the process. Participants were asked to complete the experiment as quickly and naturally as possible without compromising accuracy.

We employed a within-subjects design with three factors as *Eye-Target Distance* (1m, 2m, 4m; three levels), *Eye-Phone Distance* (25cm, 50cm; two levels) and *Corner of the Phone* (four corners; four levels). Participants had to complete 24 sessions (3 Eye-Target Distance × 2 Eye-Phone Distance × 4 Corners). We used a Latin square to balance the order of the tasks. In each session, participants performed actions 10 times (raising the phone and tapping the touchscreen), with 5 times using only left eye to observe the target (closed right eye), and vice versa. A twenty-second break was arranged after each session.

5.4 Result

We performed the occlusion rectangle estimation algorithm for all the data and calculated the pointing error between the projection of the phone's corner and the center of the crosshair. The pointing error is mainly composed of two parts, the algorithm (mainly) and the muscle jitter. In total, we collected 2880 points ($12 \times 3 \times 2 \times 4 \times 10$). We filtered outlier trials with errors exceeding $3 * \sigma$ refer to the prior related work [43], which removed 13 points. Figure 5 and Table 2 indicated the distributions of the phone and the corresponding pointing accuracy with different factors in detail. The pointing error (in angle) refers to the angle formed by the center of the rear camera, the estimated point, and the center of the crosshair. The 'Top-Left' in the figure refers to the corner under the user's perspective when the user holds the phone vertically. Correspondingly, the rear camera of the iPhone 12 Pro is located near the 'Top-Right' corner.

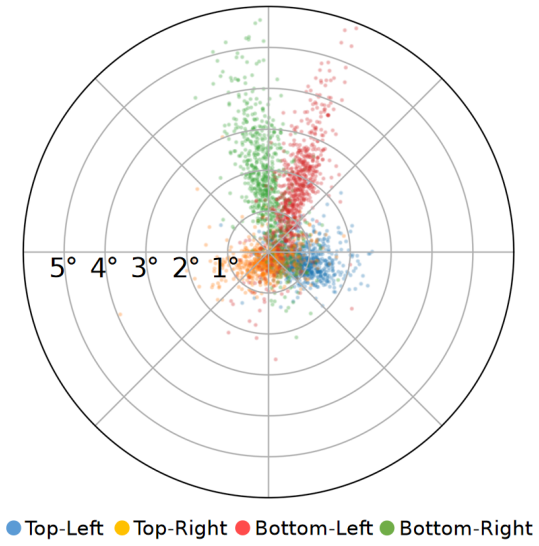


Figure 5: The distribution of the projection points of each corner of the phone with respect to the target point.

According to the analysis of Equation 2, we can find that the main factor causing the estimation error is the inaccuracy of the iris depth estimation, which corresponds to the radially outward distribution in Figure 5. This inaccuracy grows as D_x becomes larger, which explains why the 'Top-Right' corner in Table 2 is the most accurate, as it is the closest to the camera.

A Shapiro-Wilk normality test showed that the *Error* is not normally distributed ($p = .002$), so we performed a three-way ART RM-ANOVA [63]. The results showed significant effects of all *Eye-Phone Distance* ($F_{1,10} = 10.745, p = .008$), *Eye-Target Distance* ($F_{2,20} = 11.885, p < .001$), and *Corner of the Phone* ($F_{3,30} = 61.613, p < .001$) on the *Error*. We further performed the post-hoc Wilcoxon signed-rank tests on *Eye-Target Distance* and *Corner of the Phone*, respectively. For the *Eye-Target Distance*, the difference between *1m* and *2m* and between *1m* and *4m* is significant ($p < .05$). For the *Corner of the Phone*, all pairwise differences are significant ($p < .002$), except between the bottom-left corner and the bottom-right corner.

In summary, our current implementation achieved an average corner error of the occluded area estimation of 1.28°±0.96°, which was acquired with a generalized model without calibrations toward users. The average pointing error was further reduced to 0.65°±0.52° if we calibrated for each participant.

Table 2: The pointing accuracy measured by angular error (mean±SD). The three factors are the Eye-Phone Distance (Arm), Eye-Target Distance (Target), and the Corner of the Phone.

Arm	Target	Corner of the Phone				Average
		Top-Left	Top-Right	Bottom-Right	Bottom-Left	
50cm	1m	0.98°±0.12°	0.54°±0.11°	2.24°±0.27°	1.85°±0.18°	1.42°±1.02°
	2m	0.89°±0.15°	0.49°±0.06°	1.41°±0.11°	1.50°±0.21°	1.08°±0.7°
	4m	0.81°±0.09°	0.39°±0.06°	1.39°±0.16°	1.26°±0.26°	0.99°±0.74°
25cm	1m	1.17°±0.09°	0.73°±0.14°	2.18°±0.30°	1.76°±0.39°	1.48°±1.05°
	2m	1.05°±0.18°	0.60°±0.06°	1.88°±0.20°	2.10°±0.41°	1.41°±1.07°
	4m	0.92°±0.10°	0.59°±0.09°	1.69°±0.30°	1.98°±0.25°	1.31°±0.99°
Average		0.98°±0.46°	0.57°±0.38°	1.81°±1.08°	1.77°±1.01°	1.28°±0.96°

6 STUDY 3: USER EXPERIENCE

We conducted a third user study to evaluate our prototype system's efficiency, accuracy, and usability.

6.1 Candidate Methods

We chose three existing camera-based object selection techniques from previous works [43, 54] as our baseline for comparison (i.e., *Photograph*, *Snapshot*, and *Head-Gaze*). In addition to the three existing baseline methods, we propose two additional methods to extend the concept of using the phone as a real-world physical cursor (i.e., *Center Cursor* and *Corner Cursor*). Except for the first baseline of opening the camera to take pictures, the other methods do not require rendering the camera preview. The six candidate methods all use Center-to-Center Distance as the similarity measure and are described below.

M1. Photograph. The phone renders the rear camera preview in full-screen, with a red point fixed at the center of the screen to prompt the direction of the phone to the user. The object closest to the red cursor would be selected (measured by Center-to-Center Distance in section 4.6). The camera and screen preview remain open throughout the experiment. We use the camera's standard field of view (1x) instead of the wide-range camera.

M2. Snapshot. The object closest to the center of the rear camera image would be selected. This method is similar to *M1. Photograph*, but the phone's screen remains dark all the time.

M3. Head-Gaze. The user holds the phone in front of the face and uses head orientation as the virtual ray to select targets. The object with a minimum angular to the head-ray would be selected. The phone's screen remains dark all the time. This method is a reference to the implementation of WorldGaze [43].

M4. Center Cursor. Similar to occlusion, the user needs to aim at the target with the center of the screen from the first-person perspective instead of using the entire occlusion area. While there is not technically different from the *M6. Occlusion*, the user's mental model is subtly different. Unlike the *Snapshot* (M2), this method uses the eye-phone ray to aim at the target.

M5. Corner Cursor. Similar to occlusion, the user holds the phone vertically and aims at the target with the upper left corner of the screen from the first-person perspective if the participant is right-handed (use the upper right corner if the participant is left-handed). This method was chosen because using the corners to point to the target may be more accurate than using the center of the screen refer to Equation 2.

M6. Occlusion (Ours). The user raises the phone to occlude the target from the first-person perspective. The phone's screen remains dark all the time. The similarity is measured by Center-to-Center Distance instead of our more accurate distance-weighted Jaccard index to avoid better target prediction algorithms confounding the advantages of the design itself so that it can be compared with the baseline more fairly.

Assuming that the sensing techniques can be significantly improved in the future, we make some improvements to the baseline method to eliminate the limitations of the current techniques and focus on the interaction designs themselves.

For baseline M1 (*Photograph*) in the real world, waking up the camera preview and waiting for rendering has certain disadvantages in efficiency compared to the last five methods that support direct input of the intent after raising the phone. Assuming that in the future, the camera preview can be opened automatically when the user is raising the phone, we simplify the wake-up gesture in this study and render the camera preview all the time for M1 (Photograph). The user can trigger the photo by simply tapping the screen.

In order to solve the problem of inaccurate estimation of the head-gaze using the smartphone, we asked the user to wear the head-worn camera when performing the *Head-Gaze* technique (M3). The relative orientation between the camera and the user's head is calibrated so that the system can accurately measure the ground truth of head orientations. The users were informed to ignore the negative impacts of that head-worn camera while rating.

The existing algorithm is used without special modification when users use the occlusion-base method.

6.2 Participants

We recruited 13 participants (6 males, 7 females) from the local institution. The average age of participants was 23.8 years. All of them had experience with smartphones. This study took approximately 30 minutes. Each participant received $10 for compensation.

6.3 Apparatus and Platform

We conducted this study in a 7.4m length × 5.6m width × 2.8m height indoor environment, where we prepared 31 electrical devices as interactable candidates. The scene and objects are shown

in Figure 6. Each device had a unique identifier to make itself recognizable, even if its appearance was the same as others. We used iPhone 12 Pro as our experimental device on which we could build our prototype system.

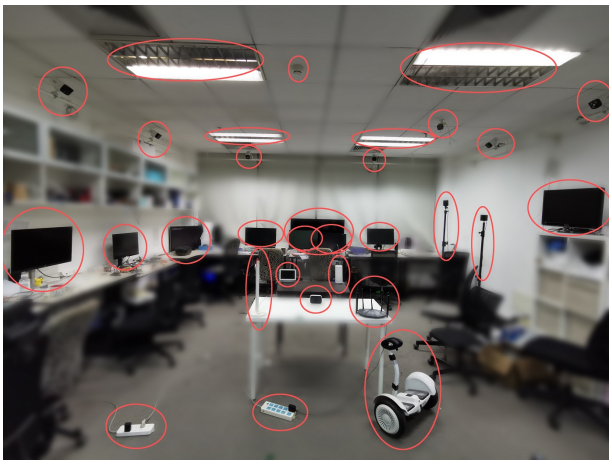


Figure 6: The indoor scene of this study from the participant's first-person perspective. The 31 selectable electronic devices are marked in the figure.

The six selection techniques employed the same neural network model for object recognition, which was trained with 30 pictures of the scene from different perspectives. For 97.39% of the tasks during the experiment, the object recognition algorithm correctly recognized the target.

6.4 Experiment Design and Procedure

The participant's task is designed to obtain the device's name. Participants need to focus on the user experience of the target selection process. First, an experimenter introduced the six different object selection methods and the interactable devices in the environment. Participants only listened to a brief how-to guide without understanding the technical details. Participants attempted each technique in the Latin square order and selected the objects freely. When they tap the screen, the phone will speak the name of the selected device via voice feedback and display its name on the screen. Besides, participants were asked to stand in a fixed position and put their hands down after each task. We measured the time cost of each selection. After participants had fully experienced each technique, we interviewed them to collect their subjective feedback. We asked them to fill out the NASA-TLX [27] questionnaire on a 7-point Likert scale and the System Usability Scale (SUS) [13] on a 5-point Likert scale as the metrics to evaluate user experience.

6.5 Result

The completion time of the process, the accuracy of target selection, and participants' subjective ranking for the six object selection techniques are shown in Figure 7.

One-way RM ANOVA were performed to compare the effect of *Methods* on *Time* and *Accuracy* with post-hoc T-tests. Friedman tests were performed to compare the effect of *Methods* on subjective scores with post-hoc Wilcoxon signed rank tests.

6.5.1 Time. Result showed significant effects of *Methods* ($F_{5,12} = 16.326, p < .001$) on *Time*. Post-hoc tests showed that our method (M6, *Occlusion*, mean=1.00s, SD=0.24s) was significantly ($p < .05$) faster than *Photograph* (M1, mean=1.65s, SD=0.43s), *Snapshot* (M2, mean=1.20s, SD=0.24s), *Head-Gaze* (M3, mean=1.59s, SD=0.46s) and *Corner Cursor* (M5, mean=1.25s, SD=0.47s). There was no significant difference between our method (M6) and *Center Cursor* (M4, mean=1.03s, SD=0.23s).

When using the *Snapshot* (M2) and *Head-Gaze* (M3), some users showed a moment of hesitation before tapping the screen. Users reported that it usually required more time to confirm the pointing direction due to a lack of visual feedback because it is easy to choose the wrong target if the user is not focused. For *Photograph* (M1), although we told the user that they can aim at the target more freely, and the algorithm will choose the closest object, most users tended to make the on-screen selection tool fall closer to the target for psychological comfort. This process of fine-tuning the phone's orientation made the *Photograph* (M1) significantly slower than *Snapshot* (M2). For the *Corner Cursor* (M5), the user felt the double vision effect which confused the users when looking at the corner of the phone. For the *Center Cursor* (M4) and *Occlusion* (M6), users reported that the visual feedback provided by the phone case allows them to easily confirm that they have correctly selected the target, so they can tap the screen to take a photo without hesitation.

6.5.2 Accuracy. Result showed significant effects of *Methods* on *Accuracy* ($F_{5,12} = 31.049, p < .001$). Post-hoc tests showed that our method (M6, mean=95.9%, SD=4.5%) was significantly ($p < .05$) more accurate than *Snapshot* (M2, mean=85.3%, SD=7.4%), *Head-Gaze* (M3, mean=65.6%, SD=14.4%) and the *Corner Cursor* (M5, mean=87.0%, SD=9.1%). Our method (M6) showed no significant difference compared to methods *Photograph* (M1, mean=99.4%, SD=1.7%) and *Center Cursor* (M4, mean=96.1%, SD=5.0%).

When using the *Snapshot* (M2) and *Head-Gaze* (M3), users complained that the lack of visual feedback restricted the accurate selection, especially for *Head-Gaze* (M3). Sometimes users think that they are pointing at the target accurately, but the ground truth is very deviated. For the *Corner Cursor* (M5), users reported that sometimes the double vision effect confused the correct phone corner locations. Users generally report that the *Photogrash* is the most accurate because they can see the cursor on the screen.

6.5.3 Subjective Feedback. Friedman tests showed that *Methods* makes a significant influence on both NASA-TLX ($p < .001$) and SUS ($p < .001$). Post-hoc tests showed that our method (M6) provides significantly ($p < .05$) lower task load (NASA-TLX) and higher system usability (SUS) than *Snapshot* (M2), *Head-Gaze* (M3), *Center Cursor* (M4) and *Corner Cursor* (M5).

By conducting interviews with users, we found that 11 out of 13 users prefer one particular technique: U3, U8, and U10 tend to use *Photograph* (M1); U2 tends to use *Center Cursor* (M4); U1, U4, U7, U9, U11, U12, and U13 tend to use *Occlusion* (M6). Users generally think that different methods have their own advantages and disadvantages, and we summarize them below.

Photograph: Users appreciated that *Photograph* is easy to understand and can accurately select dense and small targets with the visual feedback of the camera preview. Negative comments from users mainly focus on: objects on the screen look too small; indirect

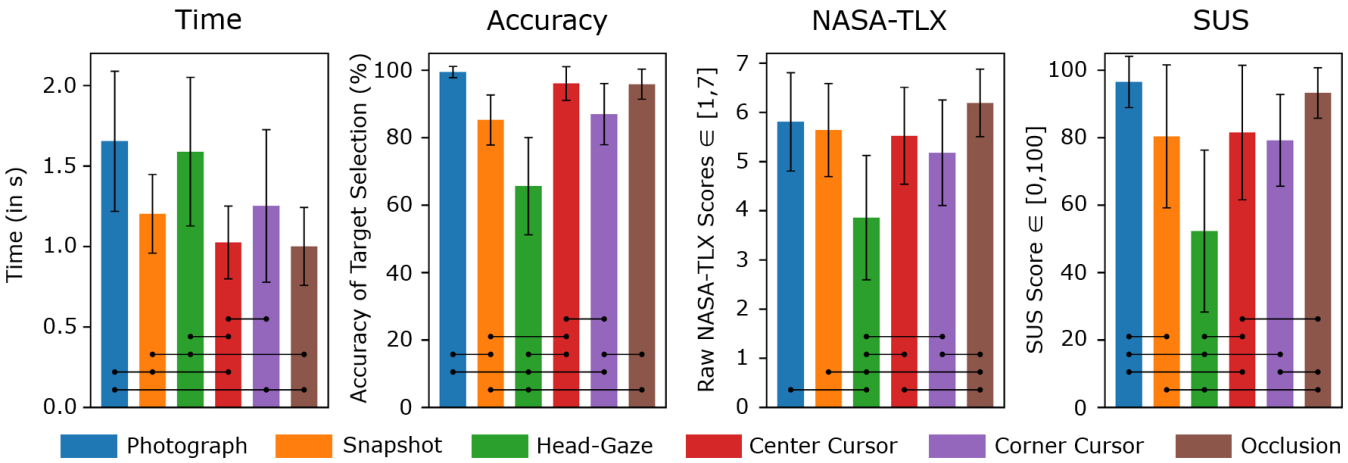


Figure 7: Completion time, accuracy of target selection, and user's subjective feedback on the six methods. The score range of the subjective feedback is from 1 to 7 for NASA-TLX and 0 to 100 for SUS (higher score represents a more positive evaluation). The standard deviation and statistical significance (p<.05) are marked in the figure.

pointing via camera preview is not as natural as user-perspective pointing.

Snapshot: The comments were polarized. A few users with a good sense of self (U5 and U6) appreciated *Snapshot* for being faster than *Photograph*. Other users were prone to making mistakes. Users, even the most skilled part, complain that the lack of visual feedback makes them less confident and more frustrated.

Head-Gaze: Inaccuracy was considered the most severe problem with *Head-Gaze*. The lack of visual feedback problem is more prominent than *Snapshot*. Many users indicated that their self-perceived head orientation was completely different from the ground truth, and it is unnatural to control the head to point precisely at the target while looking at it. We think that using eye-gaze instead of head-gaze is a better way for users to select targets, but users will not be able to look at the phone screen simultaneously in this case.

Center Cursor: Users praised the user-perspective approach for being natural and comfortable. The main problem of *Center Cursor* is that since the phone is entirely black,users feel it is difficult to estimate exactly where the center of the screen is. However, estimating the center of the screen from the outline and pointing at the target rarely selects the wrong target.

Corner Cursor: Users complained about the double-vision problem. Unlike *Center Cursor* and *Occlusion*, which can visually see the occlusion area, the *Corner Cursur* is completely split into two virtual images from the user's perspective. This makes this approach feel unnatural to the user.

Occlusion: Almost all users agreed that the concept of using the phone as a real-world physical cursor was novel and creative when they first touched it. They also felt this method is simple and comfortable after attempts. Users reported low psychological pressure and high efficiency because when they feel some overlap between the phone and the target or cannot see the target, they can easily trust that the target can be accurately selected. No users complained about the double-vision problem. Compared to other methods, one disadvantage of *Occlusion* is that it requires the user to raise the phone to a higher position. While this extra effort has a little impact during a one-shot interaction, it can cause arm fatigue during multiple consecutive uses.

Overall, our method is comparable in accuracy to *Photograph* and faster than existing baseline methods. Users consider it convenient and efficient.

7 DISCUSSION

7.1 Example Uses

Our approach provides visual context to support various vision-based context-aware interactions with in-sight objects, such as controlling the nearby ubiquitously distributed appliances [17], getting information from the objects, or performing custom actions.

For interactions with only one possible intent, such as turning on a device that is turned off, scanning the QR code, or performing a user-defined command (e.g., pet dog corresponds to opening a shopping app to buy dog food), just one wake-up gesture is enough. The iPhone supports associating a double tap on the back of the phone with a user-defined shortcut command. Using our method, the double-tap shortcut can be varied according to the selected target.

If the intent is not clear enough, multimodal interactions can be combined with gestures or voice. Gestures with different semantics can be defined for different devices. For example, we can adjust the volume by selecting the TV or smart speaker and swiping up or down on the right side of the screen. Or users can perform a select-drag-select-release gesture between devices to mirror what's playing on one screen to the other or copy configuration information between devices. Combining voice allows users to perform variable interactions, such as asking "how much is that" [43], asking the robot to "clean up that place" [32], saying "turn on" or "play music" to the device, or say "translate that" abroad on the road.

7.2 Activation Methods

Our method's role in interaction depends on the goal of the interaction task. A complete interaction process should include wake-up,

spacial object selection, and intent input. Our technique focuses on the target selecting step, but the other two parts are also essential.

Users can activate our feature using a predefined gesture (e.g., double tap the back of the phone) or wake-up word (e.g., turn on the cameras after saying "Hi, Siri"). At the same time, our rectangular ROI provides a clear spatial intention. If the semantic analysis finds that the user's voice has a clear interaction intention with the target, we have the opportunity to get rid of the wake-up words entirely and achieve the ideal of natural wake-up-free voice interaction (e.g., directly say "turn on" without "Hi, Siri"). In future work, the wake-up-free voice interaction using the occluded area as the visual cues can be studied.

7.3 Design Implications

Based on research findings from user experiments, we try to summarize some generalized design implications for 3D object selection. Accuracy, efficiency, ease of understanding and control are the four important factors we focus on that affect the user experience.

Providing appropriate visual feedback is an effective way to ensure accuracy. For *Snapshot* and *Head-Gaze*, failure to provide visual feedback can lead to lower accuracy and undermine user confidence. Our method uses the phone case to provide visual feedback, enabling one-shot interactions without the camera preview. In the future, rendering perspective-correct camera review on the screen (i.e., making the phone look like transparent glass [4, 58]) has the potential to provide more effective visual feedback for confirming the target and disambiguation for dense scenarios.

To ensure efficiency and ease of understanding and control, choosing capabilities that match the user's psychology and proficiency is desirable. We think user-perspective interaction is more natural than device-centric interaction in our application scenario if we can solve the double-vision problem. The motion control process of blocking the target is more like grabbing the target in mid-air by hand, a basic human proficiency ability. In the future, inspired by our method, we can study better solutions to the double-vision problem in various scenarios, such as VR/AR/large-screen displays, to enable user perspective interaction to work better.

7.4 Limitations and Future Work

Our prototype's object detection algorithm only recognized a predefined object collection. In the future, we should consider allowing users to register a new object by themselves, download an object's recognition model from the Internet to their smartphones, or perform object recognition by sending images to the cloud.

Using the smartphone as a rotatable and scalable rectangle occlusion box can select one of two small targets side by side by offsetting to one side. However, a post-hoc disambiguation process may be necessary when faced with three or more dense small objects and the target cannot be determined.

Our prototype's current object recognition backend can only output a rectangle bounding box for each target, which may be inappropriate for those objects with complicated form factors (e.g., a sickle-shaped object). In the future, we can use a more accurate object segmentation model to enclose the object compactly and model it with a more refined distribution instead of a bivariate normal distribution.

Since the estimation error of the depth is the main factor causing the estimation error of the occlusion rectangle, it can be improved by using the smartphones' front/rear true depth cameras (e.g., structured-light and LiDAR for depth sensing) in the future.

Most commercial smartphones are large enough to adopt the presented occlusion-based target selection technique. However, holding a small smartphone vertically (e.g., smaller than 4.7 inches) may increase selection ambiguity since there are no invisible overlapped regions from users' perspective anymore. To address this issue, users can hold the smartphone horizontally to block objects or roughly treat the phone with ghosting as an occluded area.

8 CONCLUSION

In this work, we present a novel object selection technique based on user-perspective phone occlusion which allows the user to quickly and easily interact with a large number of nearby objects. By analyzing the users' behavior in target selection, we model the users' behavior as a distance-weighted Jaccard index. Our three experiments show our method performs well in both efficiency and accuracy. Users agree that our method is convenient, accurate, efficient, and can be used as the preferred choice for one-time interactions with in-sight objects on smartphones.

ACKNOWLEDGMENTS

This work is supported by the Natural Science Foundation of China under Grant no. 62132010, and by Beijing Key Lab of Networked Multimedia, the Institute for Guo Qiang, Tsinghua University, Institute for Artificial Intelligence, Tsinghua University (THUAI), and by 2025 Key Technological Innovation Program of Ningbo City under Grant No.2022Z080.

REFERENCES

[1] Artsiom Ablavatski, Andrey Vakunov, Ivan Grishchenko, Karthik Raveendran, and Matsvei Zhdanovich. 2020. Real-time Pupil Tracking from Monocular Video for Digital Puppetry. https://doi.org/10.48550/ARXIV.2006.11341

[2] Heikki Ailisto, Lauri Pohjanheimo, Pasi Välkkynen, Esko Strömmer, Timo Tuomisto, and Ilkka Korhonen. 2006. Bridging the physical and virtual worlds by local connectivity-based physical selection. *Personal and Ubiquitous Computing* 10, 6 (2006), 333–344.

[3] Amr Alanwar, Moustafa Alzantot, Bo-Jhang Ho, Paul Martin, and Mani Srivastava. 2017. SeleCon: Scalable IoT Device Selection and Control Using Hand Gestures. In *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation* (Pittsburgh, PA, USA) *(IoTDI '17)*. Association for Computing Machinery, New York, NY, USA, 47–58. https://doi.org/10.1145/3054977.3054981

[4] Daniel Andersen, Voicu Popescu, Chengyuan Lin, Maria Eugenia Cabrera, Aditya Shanghavi, and Juan Wachs. 2016. A Hand-Held, Self-Contained Simulated Transparent Display. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*. IEEE, Piscataway, NJ, USA, 96–101. https://doi.org/10.1109/ISMAR-Adjunct.2016.0049

[5] Ferran Argelaguet and Carlos Andujar. 2013. A survey of 3D object selection techniques for virtual environments. *Computers & Graphics* 37, 3 (2013), 121–136.

[6] Huidong Bai, Li Zhang, Jing Yang, and Mark Billinghurst. 2021. Bringing full-featured mobile phone interaction into virtual reality. *Computers & Graphics* 97 (2021), 42–53. https://doi.org/10.1016/j.cag.2021.04.004

[7] Amartya Banerjee, Jesse Burstyn, Audrey Girouard, and Roel Vertegaal. 2012. MultiPoint: Comparing laser and manual pointing as remote input in large display interactions. *International Journal of Human-Computer Studies* 70, 10 (2012), 690–702. https://doi.org/10.1016/j.ijhcs.2012.05.009 Special issue on Developing, Evaluating and Deploying Multi-touch Systems.

[8] Domagoj Baričević, Cha Lee, Matthew Turk, Tobias Höllerer, and Doug A. Bowman. 2012. A hand-held AR magic lens with user-perspective rendering. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, Piscataway, NJ, USA, 197–206. https://doi.org/10.1109/ISMAR.2012.6402557

[9] Eric A. Bier, Maureen C. Stone, Ken Pier, William Buxton, and Tony D. DeRose. 1993. Toolglass and Magic Lenses: The See-through Interface. In *Proceedings*

of the 20th Annual Conference on Computer Graphics and Interactive Techniques (Anaheim, CA) *(SIGGRAPH '93)*. Association for Computing Machinery, New York, NY, USA, 73–80. https://doi.org/10.1145/166117.166126

[10] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. https://doi.org/10.48550/ARXIV.2004.10934

[11] Richard A. Bolt. 1980. "Put-That-There": Voice and Gesture at the Graphics Interface. *SIGGRAPH Comput. Graph.* 14, 3 (July 1980), 262–270. https://doi.org/10.1145/965105.807503

[12] Sebastian Boring, Dominikus Baur, Andreas Butz, Sean Gustafson, and Patrick Baudisch. 2010. *Touch Projector: Mobile Interaction through Video.* Association for Computing Machinery, New York, NY, USA, 2287–2296. https://doi.org/10.1145/1753326.1753671

[13] John Brooke et al. 1996. SUS-A quick and dirty usability scale. *Usability evaluation in industry* 189, 194 (1996), 4–7.

[14] Yuxuan Cai. 2020. *YOLObile: Real-time object detection on mobile devices via compression-compilation co-design.* Ph.D. Dissertation. Northeastern University.

[15] Kaifei Chen, Jonathan Fürst, John Kolb, Hyung-Sin Kim, Xin Jin, David E. Culler, and Randy H. Katz. 2018. SnapLink: Fast and Accurate Vision-Based Appliance Control in Large Commercial Buildings. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4, Article 129 (Jan. 2018), 27 pages. https://doi.org/10.1145/3161173

[16] Yifei Cheng, Yukang Yan, Xin Yi, Yuanchun Shi, and David Lindlbauer. 2021. SemanticAdapt: Optimization-Based Adaptation of Mixed Reality Layouts Leveraging Virtual-Physical Semantic Connections. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 282–297. https://doi.org/10.1145/3472749.3474750

[17] Adrian A. de Freitas, Michael Nebeling, Xiang 'Anthony' Chen, Junrui Yang, Akshaye Shreenithi Kirupa Karthikeyan Ranithangam, and Anind K. Dey. 2016. Snap-To-It: A User-Inspired Platform for Opportunistic Device Interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 5909–5920. https://doi.org/10.1145/2858036.2858177

[18] William Delamare, Céline Coutrix, and Laurence Nigay. 2013. Mobile Pointing Task in the Physical World: Balancing Focus and Performance While Disambiguating. In *Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Munich, Germany) *(MobileHCI '13)*. Association for Computing Machinery, New York, NY, USA, 89–98. https://doi.org/10.1145/2493190.2493232

[19] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. 2019. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE, Piscataway, NJ, USA, 6569–6578.

[20] Andrew Forsberg, Kenneth Herndon, and Robert Zeleznik. 1996. Aperture Based Selection for Immersive Virtual Environments. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology* (Seattle, Washington, USA) *(UIST '96)*. Association for Computing Machinery, New York, NY, USA, 95–96. https://doi.org/10.1145/237091.237105

[21] Ross Girshick. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*. IEEE, Piscataway, NJ, USA, 1440–1448.

[22] Taesik Gong, Hyunsung Cho, Bowon Lee, and Sung-Ju Lee. 2019. Knocker: Vibroacoustic-Based Object Recognition with Smartphones. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 3, Article 82 (sep 2019), 21 pages. https://doi.org/10.1145/3351240

[23] Google. 2020. MediaPipe Iris. Website. https://google.github.io/mediapipe/solutions/iris.

[24] Google. 2020. MediaPipe Iris: Real-time Iris Tracking & Depth Estimation. Website. https://ai.googleblog.com/2020/08/mediapipe-iris-real-time-iris-tracking.html.

[25] Tovi Grossman and Ravin Balakrishnan. 2005. *The Bubble Cursor: Enhancing Target Acquisition by Dynamic Resizing of the Cursor's Activation Area.* Association for Computing Machinery, New York, NY, USA, 281–290. https://doi.org/10.1145/1054972.1055012

[26] Tovi Grossman and Ravin Balakrishnan. 2006. The Design and Evaluation of Selection Techniques for 3D Volumetric Displays. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology* (Montreux, Switzerland) *(UIST '06)*. Association for Computing Machinery, New York, NY, USA, 3–12. https://doi.org/10.1145/1166253.1166257

[27] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index) : Results of Empirical and Theoretical Research. *Advances in Psychology* 52, 6 (1988), 139–183.

[28] Jeremy Hartmann and Daniel Vogel. 2018. An Evaluation of Mobile Phone Pointing in Spatial Augmented Reality. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI EA '18)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3170427.3188535

[29] Ken Hinckley, Randy Pausch, John C. Goble, and Neal F. Kassell. 1994. A Survey of Design Issues in Spatial Input. In *Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology* (Marina del Rey, California, USA) *(UIST '94)*. Association for Computing Machinery, New York, NY, USA, 213–222. https://doi.org/10.1145/192426.192501

[30] Rachel Huang, Jonathan Pedoeem, and Cuixian Chen. 2018. YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers. In *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, Piscataway, NJ, USA, 2503–2510. https://doi.org/10.1109/BigData.2018.8621865

[31] Ltd. Huawei Device Co. 2021. This Button Can Do More Things Than Expected. Website. https://consumer.huawei.com/za/support/article-list/article-detail/en-us15759678/.

[32] Runchang Kang, Anhong Guo, Gierad Laput, Yang Li, and Xiang 'Anthony' Chen. 2019. Minuet: Multimodal Interaction with an Internet of Things. In *Symposium on Spatial User Interaction* (New Orleans, LA, USA) *(SUI '19)*. Association for Computing Machinery, New York, NY, USA, Article 2, 10 pages. https://doi.org/10.1145/3357251.3357581

[33] Mohamed Khamis, Florian Alt, and Andreas Bulling. 2018. The Past, Present, and Future of Gaze-Enabled Handheld Mobile Devices: Survey and Lessons Learned. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Barcelona, Spain) *(MobileHCI '18)*. Association for Computing Machinery, New York, NY, USA, Article 38, 17 pages. https://doi.org/10.1145/3229434.3229452

[34] Mikko Kytö, Barrett Ens, Thammathip Piumsomboon, Gun A. Lee, and Mark Billinghurst. 2018. Pinpointing: Precise Head- and Eye-Based Target Selection for Augmented Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3173574.3173655

[35] Cha Lee, Scott Bonebrake, Tobias Hollerer, and Doug A. Bowman. 2009. A replication study testing the validity of AR simulation in VR for controlled experiments. In *2009 8th IEEE International Symposium on Mixed and Augmented Reality*. IEEE, Piscataway, NJ, USA, 203–204. https://doi.org/10.1109/ISMAR.2009.5336464

[36] SangYoon Lee, Jinseok Seo, Gerard Jounghyun Kim, and Chan-Mo Park. 2003. Evaluation of pointing techniques for ray casting selection in virtual environments. In *Third International Conference on Virtual Reality and Its Application in Industry*, Zhigeng Pan and Jiaoying Shi (Eds.), Vol. 4756. International Society for Optics and Photonics, SPIE, Bellingham, WA, USA, 38 – 44. https://doi.org/10.1117/12.497665

[37] Jiandong Liang and Mark Green. 1993. Geometric modeling using six degrees of freedom input devices. , 217–222 pages.

[38] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, Piscataway, NJ, USA, 2117–2125.

[39] Julian Looser, Mark Billinghurst, and Andy Cockburn. 2004. Through the Looking Glass: The Use of Lenses as an Interface Tool for Augmented Reality Interfaces. In *Proceedings of the 2nd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia* (Singapore) *(GRAPHITE '04)*. Association for Computing Machinery, New York, NY, USA, 204–211. https://doi.org/10.1145/988834.988870

[40] Julian Looser, Mark Billinghurst, Raphaël Grasset, and Andy Cockburn. 2007. An Evaluation of Virtual Lenses for Object Selection in Augmented Reality. In *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia* (Perth, Australia) *(GRAPHITE '07)*. Association for Computing Machinery, New York, NY, USA, 203–210. https://doi.org/10.1145/1321261.1321297

[41] 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)*. IEEE, Piscataway, NJ, USA, 35–43.

[42] Diako Mardanbegi, Benedikt Mayer, Ken Pfeuffer, Shahram Jalaliniya, Hans Gellersen, and Alexander Perzl. 2019. EyeSeeThrough: Unifying Tool Selection and Application in Virtual Environments. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, Piscataway, NJ, USA, 474–483. https://doi.org/10.1109/VR.2019.8797988

[43] Sven Mayer, Gierad Laput, and Chris Harrison. 2020. Enhancing Mobile Voice Assistants with WorldGaze. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3313831.3376479

[44] Sven Mayer, Valentin Schwind, Robin Schweigert, and Niels Henze. 2018. The Effect of Offset Correction and Cursor on Mid-Air Pointing in Real and Virtual Environments. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3173574.3174227

[45] Sven Mayer, Katrin Wolf, Stefan Schneegass, and Niels Henze. 2015. Modeling Distant Pointing for Compensating Systematic Displacements. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul,

Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 4165–4168. https://doi.org/10.1145/2702123.2702332

[46] Mark R Mine. 1995. Virtual environment interaction techniques.

[47] Kai Nickel and Rainer Stiefelhagen. 2003. Pointing Gesture Recognition Based on 3D-Tracking of Face, Hands and Head Orientation. In *Proceedings of the 5th International Conference on Multimodal Interfaces* (Vancouver, British Columbia, Canada) *(ICMI '03)*. Association for Computing Machinery, New York, NY, USA, 140–146. https://doi.org/10.1145/958432.958460

[48] Dan R. Olsen and Travis Nielsen. 2001. Laser Pointer Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Seattle, Washington, USA) *(CHI '01)*. Association for Computing Machinery, New York, NY, USA, 17–22. https://doi.org/10.1145/365024.365030

[49] Jeffrey S. Pierce, Andrew S. Forsberg, Matthew J. Conway, Seung Hong, Robert C. Zeleznik, and Mark R. Mine. 1997. Image Plane Interaction Techniques in 3D Immersive Environments. In *Proceedings of the 1997 Symposium on Interactive 3D Graphics* (Providence, Rhode Island, USA) *(I3D '97)*. Association for Computing Machinery, New York, NY, USA, 39–ff. https://doi.org/10.1145/253284.253303

[50] Yue Qin, Chun Yu, Zhaoheng Li, Mingyuan Zhong, Yukang Yan, and Yuanchun Shi. 2021. ProxiMic: Convenient Voice Activation via Close-to-Mic Speech Detected by a Single Microphone. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 8, 12 pages. https://doi.org/10.1145/3411764.3445687

[51] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE, Piscataway, NJ, USA, 779–788.

[52] Jie Ren, Yueting Weng, Chengchi Zhou, Chun Yu, and Yuanchun Shi. 2020. Understanding Window Management Interactions in AR Headset + Smartphone Interface. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '20)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/3334480.3382812

[53] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* 28 (2015), 91–99.

[54] Michael Rohs and Antti Oulasvirta. 2008. Target Acquisition with Camera Phones When Used as Magic Lenses. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*. Association for Computing Machinery, New York, NY, USA, 1409–1418. https://doi.org/10.1145/1357054.1357275

[55] Michael Rohs, Antti Oulasvirta, and Tiia Suomalainen. 2011. *Interaction with Magic Lenses: Real-World Validation of a Fitts' Law Model.* Association for Computing Machinery, New York, NY, USA, 2725–2728. https://doi.org/10.1145/1978942.1979343

[56] Robin Schweigert, Valentin Schwind, and Sven Mayer. 2019. EyePointing: A Gaze-Based Selection Technique. In *Proceedings of Mensch Und Computer 2019* (Hamburg, Germany) *(MuC'19)*. Association for Computing Machinery, New York, NY, USA, 719–723. https://doi.org/10.1145/3340764.3344897

[57] Ke Sun, Chun Yu, and Yuanchun Shi. 2019. Exploring Low-Occlusion Qwerty Soft Keyboard Using Spatial Landmarks. *ACM Trans. Comput.-Hum. Interact.* 26, 4, Article 20 (June 2019), 33 pages. https://doi.org/10.1145/3318141

[58] Yuko Unuma, Takehiro Niikura, and Takashi Komuro. 2014. See-through Mobile AR System for Natural 3D Interaction. In *Proceedings of the Companion Publication of the 19th International Conference on Intelligent User Interfaces* (Haifa, Israel) *(IUI Companion '14)*. Association for Computing Machinery, New York, NY, USA, 17–20. https://doi.org/10.1145/2559184.2559198

[59] Pasi Välkkynen, Marketta Niemelä, and Timo Tuomisto. 2006. Evaluating Touching and Pointing with a Mobile Terminal for Physical Browsing. In *Proceedings of the 4th Nordic Conference on Human-Computer Interaction: Changing Roles* (Oslo, Norway) *(NordiCHI '06)*. Association for Computing Machinery, New York, NY, USA, 28–37. https://doi.org/10.1145/1182475.1182479

[60] Klen Čopič Pucihar, Paul Coulton, and Jason Alexander. 2013. Evaluating Dual-View Perceptual Issues in Handheld Augmented Reality: Device vs. User Perspective Rendering. In *Proceedings of the 15th ACM on International Conference on Multimodal Interaction* (Sydney, Australia) *(ICMI '13)*. Association for Computing Machinery, New York, NY, USA, 381–388. https://doi.org/10.1145/2522848.2522885

[61] John Viega, Matthew J. Conway, George Williams, and Randy Pausch. 1996. 3D Magic Lenses. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology* (Seattle, Washington, USA) *(UIST '96)*. Association for Computing Machinery, New York, NY, USA, 51–58. https://doi.org/10.1145/237091.237098

[62] Thomas Vincent, Laurence Nigay, and Takeshi Kurata. 2013. Precise Pointing Techniques for Handheld Augmented Reality. In *Human-Computer Interaction – INTERACT 2013*, Paula Kotzé, Gary Marsden, Gitte Lindgaard, Janet Wesson, and Marco Winckler (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 122–139.

[63] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011. *The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only Anova Procedures.* Association for Computing Machinery, New York, NY, USA, 143–146. https://doi.org/10.1145/1978942.1978963

[64] Robert Xiao, Gierad Laput, Yang Zhang, and Chris Harrison. 2017. *Deus EM Machina: On-Touch Contextual Functionality for Smart IoT Appliances.* Association for Computing Machinery, New York, NY, USA, 4000–4008. https://doi.org/10.1145/3025453.3025828

[65] Yukang Yan, Yingtian Shi, Chun Yu, and Yuanchun Shi. 2020. HeadCross: Exploring Head-Based Crossing Selection on Head-Mounted Displays. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 1, Article 35 (March 2020), 22 pages. https://doi.org/10.1145/3380983

[66] Yukang Yan, Chun Yu, Yingtian Shi, and Minxing Xie. 2019. PrivateTalk: Activating Voice Input with Hand-On-Mouth Gesture Detected by Bluetooth Earphones. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 1013–1020. https://doi.org/10.1145/3332165.3347950

[67] Zhican Yang, Chun Yu, Fengshi Zheng, and Yuanchun Shi. 2019. ProxiTalk: Activate Speech Input by Bringing Smartphone to the Mouth. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 3, Article 118 (Sept. 2019), 25 pages. https://doi.org/10.1145/3351276

[68] Jibin Yin, Chengyao Fu, Xiangliang Zhang, and Tao Liu. 2019. Precise Target Selection Techniques in Handheld Augmented Reality Interfaces. *IEEE Access* 7 (2019), 17663–17674. https://doi.org/10.1109/ACCESS.2019.2895219

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

[70] Ben Zhang, Yu-Hsiang Chen, Claire Tuna, Achal Dave, Yang Li, Edward Lee, and Björn Hartmann. 2014. HOBS: Head Orientation-Based Selection in Physical Spaces. In *Proceedings of the 2nd ACM Symposium on Spatial User Interaction* (Honolulu, Hawaii, USA) *(SUI '14)*. Association for Computing Machinery, New York, NY, USA, 17–25. https://doi.org/10.1145/2659766.2659773