Portal-ble: Intuitive Free-hand Manipulation in Unbounded Smartphone-based Augmented Reality

Jing Qian, Jiaju Ma, Xiangyu Li*, Benjamin Attal, Haoming Lai, James Tompkin, John F. Hughes, Jeff Huang

Brown University, Providence RI, USA
*Southeast University, Nanjing, China

ABSTRACT

Smartphone augmented reality (AR) lets users interact with physical and virtual spaces simultaneously. With 3D hand tracking, smartphones become apparatus to grab and move virtual objects directly. Based on design considerations for interaction, mobility, and object appearance and physics, we implemented a prototype for portable 3D hand tracking using a smartphone, a Leap Motion controller, and a computation unit. Following an experience prototyping procedure, 12 researchers used the prototype to help explore usability issues and define the design space. We identified issues in perception (moving to the object, reaching for the object), manipulation (successfully grabbing and orienting the object), and behavioral understanding (knowing how to use the smartphone as a viewport). To overcome these issues, we designed object-based feedback and accommodation mechanisms and studied their perceptual and behavioral effects via two tasks: picking up distant objects, and assembling a virtual house from blocks. Our mechanisms enabled significantly faster and more successful user interaction than the initial prototype in picking up and manipulating stationary and moving objects, with a lower cognitive load and greater user preference. The resulting system—Portal-ble—improves user intuition and aids free-hand interactions in mobile situations.

Author Keywords

Smartphone, augmented reality, 3D hand manipulation.

CCS Concepts

•**Human-centered computing** → **Mixed/augmented reality; Gestural input;** •**Computing methodologies** → **Perception;**

INTRODUCTION

Our hands are an interface to the physical environments around us: we pick up cups, turn doorknobs, and carry groceries. In augmented reality (AR), we might wish to use our hands to interact with *virtual* objects within a physical space, for which intuitive interaction methods must be designed. How can we accomplish this for everyday devices like smartphones? Prior work provides evidence that direct free-hand manipulation is a potentially useful interaction method for smartphone AR [6, 23, 26, 29, 51]. However, despite recent improvements in 3D hand tracking, there are still perceptual, ergonomic, and usability challenges with smartphone AR technology. For instance, there are known challenges with depth perception, with the smartphone form factor, and with the view-angle offset between our eyes and the smartphone's rear camera [27]. These can all affect our ability to manipulate virtual objects with our hands and require consideration when designing interaction methods.

Further, smartphone AR allows us to navigate to and interact with virtual objects at different physical locations, providing potentially *unbounded* mobility. Existing 3D free-hand systems were, however, tested in stationary conditions which did not offer the mobility afforded by smartphone AR systems. As such, how people use free-hand interactions in a mobile setting is under-explored and necessary to consider.

We investigate free-hand manipulation issues in smartphone AR with the goal of creating more intuitive interaction methods. First, we built an initial prototype: a smartphone with a Leap Motion hand tracker fixed to its back, and a portable computation unit with a trained hand gesture classification model. Next, we used an experience prototyping procedure to help identify usability issues related to depth perception, manipulation, and a lack of common gestural behaviors when interacting with virtual objects in large spaces.

Understanding these issues informed the design of our system—Portal-ble—with visual, auditory, and haptic feedback mechanisms which accommodate user behaviors in free-hand AR interactions. An empirical study evaluating the efficacy of Portal-ble showed that these methods significantly improve both the perception of virtual objects' spatial locations and the user success rate in manipulating virtual objects. Further, it showed that the system helps users establish mental models for more efficient free-hand interaction in smartphone AR.

The contributions of this work are:

1. The identification of usability, perception, and manipulation challenges in 3D free-hand AR interactions through an experience prototyping procedure.
2. The Portal-ble system to meet these challenges through feedback and accommodation for visual, auditory, and haptic senses, and the evaluation of the system via a user study.

We release our initial system prototype and the final Portal-ble system as open source for experiment reproduction, practical use, and extension. `https://portalble.cs.brown.edu/`

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
UIST '19, October 20-23, 2019, New Orleans, LA, USA.
© 2019 Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-6816-2/19/10 ...$15.00.
https://doi.org/10.1145/3332165.3347904

RELATED WORK

AR on Mobile Devices

Due to the shrinking size of high-powered computing hardware, we can now interact with AR content on mobile devices. Mobile AR in the 1990s required a user to carry a set of equipment with a personal computer on their back [15, 49]. Systems like ARQuake [48] and Exploring MARS [22] enabled users to experience a world with mixed physical and virtual objects while navigating through a large space.

In the 2000s, the emergence of tablets and smartphone devices made AR lightweight and portable, enabling new potential applications. Reitmayr and Drummond used a handheld tablet to recognize and track building features in an urban environment [38]; Iris et al. visually revived *Heinzelmannchen*—legendary elves from a German folk tale—in front of users' eyes with the mobile AR game TimeWarp [19]. Researchers have also implemented mobile AR systems for medical surgery [8], social collaboration [1], tourism [12], and architecture [48], to name a few. Today, the popularity of mobile AR is demonstrated in games like *Pokemon Go*, which are available to anyone with a smartphone. However, interaction in smartphone AR is typically limited to touchscreen button presses and gestures.

Free-hand Direct Manipulation in Handheld AR

One way to improve smartphone AR interactivity and preserve portability is to allow intuitive hand-based interaction. On a handheld device, marker-less free-hand interaction in 3D can be achieved through incorporating additional components for computer vision techniques [28, 39, 46], depth sensing [5, 32, 37, 44], or deep convolutional neural network (CNN) based models [31, 33]. Song et al. [46] use vision-based methods to determine hand pose in 2D, namely background subtraction with skin-color detection and nearest neighbor search. They further simulate hand depth based on the device's position, but lack the ability to measure the physical 3D depth. ManoMotion [52] uses a large data set of hand images to classify 3D hand gestures to better predict when users pick, move, drop, and rotate virtual objects using a smartphone.

Deep CNN models allow robust marker-less 3D hand interfaces through a single RGB camera where vision-based methods become unreliable. They provide robust hand posture estimations, as well as finger tracking [31]. However, few systems can run natively on smartphones without limiting tracking capability [25].

Depth sensors such as Microsoft Kinect or Leap Motion facilitate hand tracking by measuring the physical depth of hands and fingers. The BeThere system installed a depth-sensor next to a smartphone to map 3D hand gestures into AR scenes [44]. Likewise, Kim et al. used a Leap Motion on a tablet to study the efficacy of 3D hand tracking in mobile AR [26]. Our system uses depth-based 3D hand tracking on the smartphone for free-hand manipulation. This approach forms the basis for our usability experiments and accommodation mechanisms, which helps to improve the practical uses of smartphone-based free-hand manipulation.

Usability Issues with Free-hand Interactions

Prior work identified usability issues in smartphone AR environments, including missing depth perception [6, 17], difficulties in manipulating 3D objects [26], and viewport constraints [41]. Despite not being directly documented in studies using free-hand manipulations on smartphones, fatigue from the gorilla arm effect might also be a problem [4]. Multiple free-hand manipulation techniques have been explored in HMD and VR environments [34, 35, 36, 43], but not specifically for smartphones: The unique view angle offset and monoscopic displays on handheld devices [27] suggests that these techniques may not be directly applicable to smartphones. User Experience Guidelines from Leap Motion [30] provide suggestions for improving depth perception and overall free-hand interactions; however, these guidelines have not been publicly evaluated in a smartphone context. We design feedback and accommodation mechanisms to improve smartphone free-hand interaction experiences and evaluate their impact.

Manipulating Virtual Objects

Various strategies have been explored for hand-based manipulation of virtual objects in AR applications. These include gesture-based direct and indirect manipulation of objects [23, 42, 50], direct physically-based manipulation [13, 20], and methods which combine both of these approaches [2, 10, 18, 26, 34]. These strategies have been employed for AR applications on HMDs, tablets, mobile devices, and custom devices [20]. Further, Buchmann et al. found that adding occlusion and haptic feedback improved the ease of use of direct hand manipulation [10]. Our system explores similar strategies for smartphones and focuses on adaptive manipulation for different users.

INITIAL SMARTPHONE AR PROTOTYPE

To assess the experience of performing free-hand interactions on smartphones, we developed an initial prototype system for an experience prototyping procedure. This was informed by recent literature on mobility, physics, interaction spaces, and virtual object appearance in AR, as a user should not notice or be hampered by these considerations in a well-designed system.

Design Considerations

Mobility: We must consider the trade-off between system performance and ergonomics. Carrying large or heavy components is awkward and fatiguing (e.g., gorilla arm [21]), but small devices may overheat or have insufficient compute to track hands and render AR scenes. Thus, we design our prototype to add on to existing smartphones, to use light materials and components, and to be easily detachable from the smartphone. There is no reliance on external markers or internet access. This leads to a self-contained portable system which permits full mobility.

Physics: Seo and Lee [43] show how direct manipulation can enable intuitive interaction between virtual and physical spaces. However, this requires changing the traditional physical behavior of virtual objects: unlike with physical objects, it is easy for users to put their hands *through* virtual objects. This might force them to fly away if traditional physics were calculated [20]. Similarly, it is easy to accidentally drop an object without any tactile feedback from the object itself.

Interaction Spaces: A fully mobile smartphone with free-hand tracking requires the coordination of two interaction spaces: the first is the area within arm's reach behind the smartphone,

Figure 1. A: The initial prototype in use, which has the same hardware setup as the Portal-ble system; B: Portal-ble in use; C: The hardware components of the initial prototype.

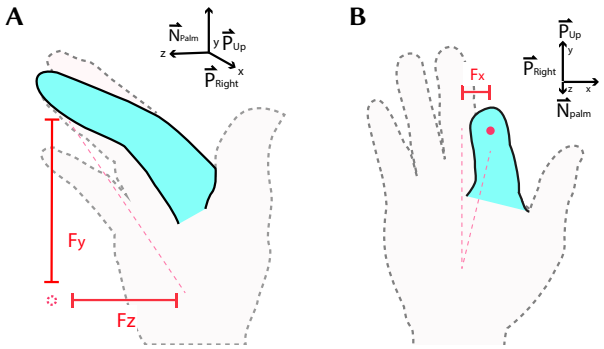

Figure 2. Each fingertip is represented by three features (F_x,F_y,F_z). F_z is a projection formed with the fingertip to palm center and a normal vector to the palm ($\vec{N}_{Palm}$). The remaining two features are calculated similarly with a projection from the palm vector ($\vec{N}_{Palm}$) and each of the two local vectors $\vec{P}_{Up}$ and $\vec{P}_{Right}$.

where users perform free-hand direct manipulation; the second is the physical environment around the user in which they navigate. This second space does not require active hand tracking but can be made interactive when the user moves into it. The division of space in this manner enables portability for users while retaining direct manipulation capabilities with virtual objects even outside of arm's reach.

Virtual Object Appearance: This plays a key role in helping users to understand spatial relationships [47]. Including such cues requires consideration of the lighting conditions and geometries in the smartphone's surroundings. For example, virtual objects in a well-lit, blue room should be brighter and tinted by the blue ambient light as opposed to the dimmer presentation of those same objects when they reside in a night-time scene.

Initial Prototype

Given these considerations, next we describe our initial prototype. We begin with any modern Android smartphone; in our case, a Samsung S9+. This device provides AR capabilities via Google's ARCore SDK, which performs both spatial tracking to localize the smartphone and environment lighting estimation to improve the realism of virtual object appearance. We develop our application in Unity, for which ARCore has compatible assets to define and render virtual objects.

To detect 3D hand positions and rotations, we use a Leap Motion: a dual wide-angle stereo depth-sensing camera system. We attach the Leap Motion to our smartphone using a 3D-printed mount with suction cups (Figure 1). This setup allows for simultaneous spatial and 3D hand tracking in the same coordinate system if we can combine the two camera systems via *intrinsic* and *extrinsic* calibrations. ARCore and Leap Motion both provide intrinsic calibrations and define their object positions in metric space. For extrinsic calibration, we measure the physical distance between the smartphone's back camera and the midpoint of Leap Motion's two front cameras.

Leap Motion does not directly support smartphones as its software driver only works on desktop-class operating systems. As such, we bypass this technical issue by introducing a wearable computation unit as a data relay (Intel CS325 compute stick and a 22,000 mAh battery). It wirelessly forwards the Leap Motion's hand tracking data to the smartphone via a mobile hotspot. The compute stick runs a C++ WebSocket server to transmit the Leap Motion hand data, which is a serialized string with seven variables: a hand label, the tracked palm's position, speed, normal vector and orientation, and finger joints' position and orientation. On the smartphone side, a WebSocket client receives and stores data for gesture predictions. This arrangement achieves a latency of 25 ms.

To predict 3D free-hand gestures, Leap Motion provides an Interaction Engine as part of the Orion 4.0 framework [14]. However, this is only possible when a Unity scene runs on a 64-bit Windows OS computer[1]. Therefore, we trained a 3D gestural model using a Support Vector Machine (SVM) to classify five gestures: *pinch*, *fist*, *palm*, *index*, and *idle*. The SVM gesture classifier is trained on motion-invariant projection features to ensure that the same outcomes are produced when users interact with virtual objects from different perspectives with the same gesture. These features are calculated by measuring the projected distance from the user's fingertips to their palm's center (Figure 2).

To train the model, we collected a total of 16,000 data samples with 30 features each. These data are split into training and testing sets in a ratio of 7:3. The SVM model used a 10-fold cross validation (SVM kernel = histogram intersection, C = 20). The samples were collected from 2 people who performed 5 gestures in front of the Leap Motion tracker at a range of 20 cm. The collected data was normalized to accommodate different hand sizes. Overall, our model achieved 98% accuracy in predicting those gestures. This was sufficient for our later investigation: from the qualitative experiences we observed in the user study, no one's experience was hampered by the gesture prediction.

Finally, with hand tracking and gesture detection support on the smartphone, we created a virtual 3D hand model to re-enact

[1] As we have now added a compute stick capable of running 64-bit Windows, one might conceivably run a 'mirror' Unity scene on the unit to use Interaction Engine's gesture prediction. This would require synchronizing all virtual objects between the smartphone and computational unit scenes; instead, we trained our own gesture classifier.

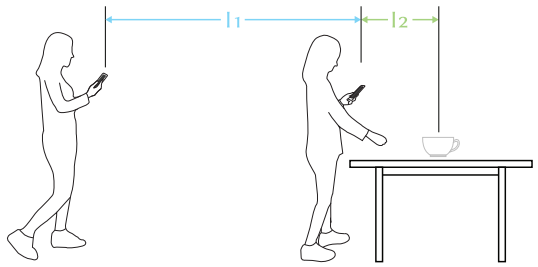

Figure 3. I1 denotes the distance from users' current location to a range where the objects are within their reach (I2).

the hand data sent from Leap Motion. The hand model is automatically normalized to the size of users' hands when the prototype initializes. During run-time, this hand model moves and rotates in real-time with the user's hand, and is calibrated to visually overlap with it. When the real hand reaches a virtual object, the overlapping virtual hand also reaches the target and can trigger interaction events. In sum, our prototype provides 3D hand tracking in AR with mobility in everyday spaces.

EXPERIENCE PROTOTYPING

We wish to reveal crucial underlying usability issues which will inform the general use of free-hand AR systems on smartphones. We accomplish this via experience prototyping: a methodology which identifies usability issues by presenting users with early prototypes [9]. Through this process, we can discover which questions are important to ask; in our case, questions such as "Are users challenged when picking up distant objects?" and "Can users comfortably and accurately interact with moving objects?" Additionally, experience prototyping allows us to gather first-hand accounts of usability and to group identified issues into categories. This approach is particularly suited for AR interaction which is hard to imagine without experiencing it.

We asked 12 student researchers to design tasks with our smartphone prototype. Each researcher was given general guidelines from which to design two tasks: one *content manipulation* task and one *content creation* task. We define *content manipulation* as interactions such as selection, translation, and confirmation via direct manipulation. The designed tasks included throwing darts, ten-pin bowling, stacking cubes, and moving objects across the room. We define *content creation* as various free-hand mid-air drawings. The designed tasks included drawing cubes, drawing lines across the room, and participants drawing their names.

To test their experiences, each of the 12 student researchers recruited two participants (24 total). Tests occurred in a 3×8 m room, and each session was video recorded. To begin, student researchers explained the concept of free-hand direct manipulation, and asked them to think aloud. Then, during the tasks, student researchers noted any user issues and recorded participant feedback, e.g., a participant trying to reach for a virtual object multiple times. Next, student researchers interviewed each participant after the session about their overall experience to identify what they did or did not enjoy about the prototype, and what specific problems were encountered. Finally, we categorized the discovered perception, manipulation, and behavior issues that impeded user ability to manipulate and create content.

Perception Issues

Over half of the participants ($N = 16$) could not understand the spatial location of their hands relative to the virtual objects displayed on the smartphone screen. For example, one participant said "I really can't tell how close or how far [away] it is;" another participant thought the object appeared farther away than it actually was. These issues manifest in two ways (Figure 3):

I1: Participants hesitated when approaching distant virtual objects. While participants could move towards virtual objects easily, they found it challenging to estimate the remaining distance to virtual objects. Participants stated aloud, "I don't know how far away that is" and "why can't I grab it?" Some stood still and tried to grab virtual objects at distances of greater than 1 m. Even after student researchers prompted them to move forward, most participants still hesitated. Similar behavior was observed in people with reduced depth perception [11].

I2: Participants were uncertain how to map the spatial distance to virtual objects. Participants were unsure how far they should extend their hands, even when the virtual object was within their reach. Several participants kept moving their hand closer to the smartphone since they thought the object was near it. During mid-air drawing tasks, participants complained that they could not align their drawing strokes to create a cohesive image.

Manipulation Issues

I3: Participants experienced difficulties picking up and releasing virtual objects, aggravated by unintentional dropping. Six participants reported that they had unintentionally dropped virtual objects during manipulation, and described these events as "annoying" and "interruptive" to the overall experience. Unlike physical objects, virtual objects are difficult to grab with precision [20]. This is due to complications which arise from coordinating the gesturing hand and smartphone, tracking limitations [26], and lack of depth perception.

We noticed two additional issues which were not included in our design considerations. First, tracking noise caused detected hand positions to fluctuate, which made it hard to decide when the hand is inside a virtual object's collision detection region. Second, the seemingly-natural pinch gesture caused frustration in picking up and releasing virtual objects. Watching video replays and analyzing notes revealed three causes: 1) a recessed grabbing behavior, e.g., unknowingly moving their fingers outside of the detection range (Figure 4); 2) an occluded grabbing behavior, e.g., pinching while the back of the hand is facing the smartphone camera; and 3) tracking issues from stray infrared interference, e.g., from fluorescent lamps.

Behavioral issues

I4: Participants lack a general mental model for free-hand direct manipulation on the smartphone. We found that participants adopted different strategies to directly manipulate virtual objects. To find the right interaction depth, some participants moved their smartphone, others moved their hands while keeping their smartphone still, and the rest simultaneously coordinated the movement of their smartphone and their hands. This process often led to participants' hands moving into untrackable regions [23], resulting in extra physical and mental effort.

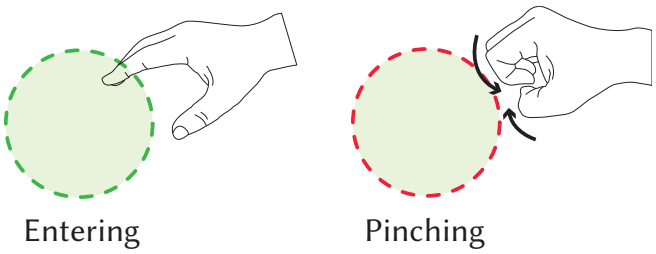

Figure 4. ***Recessed Grab:*** **When grabbing an object, the hand closes around a point which is away from the object's surface. A hand which was originally within the gestural detection region (left) is no longer inside the region after the interaction (right), resulting in a failed grab.**

Participants also had difficulty distinguishing interaction states during manipulation. Some participants repeatedly performed the pinch gesture even after grabbing an object, and two had reported that they were unsure whether they had the virtual object in hand. These issues confused them and prolonged the process of forming a mental model for free-hand interactions.

PORTAL-BLE

From the discovered usability issues, we designed and implemented the Portal-ble system to accommodate perceptions of distance, reach, and interaction state, to accommodate manipulation, and to accommodate behavior via helping users build mental models of free-hand manipulation in smartphone AR.

Accommodation for Perception

Successful free-hand manipulation with unbounded mobility requires users to overcome the issues found in **I1** and **I2**. To aid users, we consider three accommodation mechanisms: 1) cues to contextualize spatial distances, 2) indicators for when the object is within reach and 3) feedback for *default*, *hover*, and *pinch* interaction states.

For these purposes we designed a progress wheel visualization, highlight shading, haptic feedback, and distance-based sound. **Progress wheel** and **sound** were dedicated to improving spatial estimation (**I1**) by offering indicators when an object is within interaction range. Once a user is near a virtual object, **highlight** and **haptics** are designed to gauge the hand's depth and provide interaction feedback (**I2**). The design and implementation of these accommodation mechanisms is described below.

Progress Wheel

This is a two-stage non-linear visual feedback mechanism which responds to different distance intervals. In the AR environment, each virtual object displays a progress wheel which fills itself as the user approaches an object (see Figure 5). Because this action could come from any direction or distance, this visualization incorporates a non-linear sensitivity to users' distances from the virtual objects. The progress wheel is green when the objects are beyond reach and turns to blue when a user's distance from an object decreases to less than 75cm, which is the average length of a human arm ($V_{armLength}$).

Our implementation used the distance between the smartphone and the virtual object as the fill color saturation (V_{fill}) and transparency ($V_{transparency}$) values of the progress wheel. Three variables are considered: the distance from camera to object

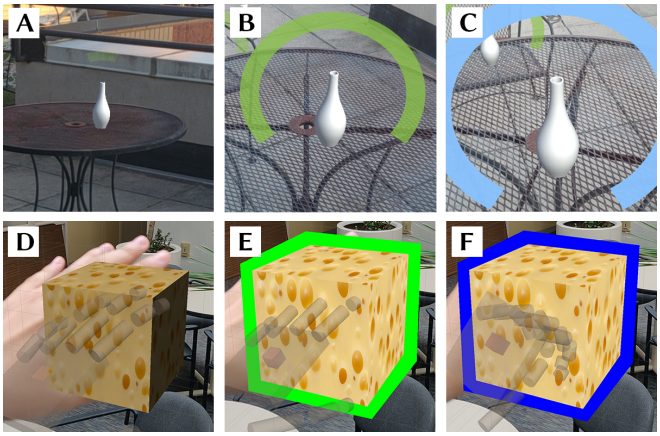

Figure 5. Two types of visual feedback for Portal-ble's free-hand interaction. ***Top:*** **A** ***progress wheel*** **demonstrates when a user A) is far away; B) getting closer, C) within reach of the object.** ***Bottom:*** **Different highlights represent that a user's hand is D) nearby, E) hovering, or F) grabbing.**

($D_{cameraToObj}$), the distance from the center of the virtual object to its surface ($V_{padding}$), and the maximum detection range D_{max}. We apply a quadratic curve for a non-linear weighting:

$$V_{transparency} = V_{fill} = \left(\frac{D_{cameraToObj} + V_{padding}}{D_{max}} \right)^2 \quad (1)$$

The progress wheel's color begins to change from green to blue when $D_{cameraToObj}$ is less than $V_{armLength}$. When $D_{cameraToObj} = V_{armLength}/2$, its color fully changes to blue, indicating that the object is within reach.

Highlight

We designed a visual signal to offer feedback when a user reaches a virtual object. This design draws a green contour line around virtual objects when they are within users' reach. The contour line turns blue when the objects are being grabbed, reflecting a change in interaction state. This feedback helps users to distinguish when the hand is entering, grabbing, holding, or releasing a virtual object (Figure 5).

Sound

Audio cues allow users to estimate physical distance based on the sound they hear when they approach the object. For example, sounds of varying pitch can be used to identify spatial distance. Audio has been shown to be an effective cue for understanding location in spatial tasks [45].

During our design process, one challenge arose in avoiding simultaneous sound generation from multiple virtual objects. A ray casting technique [40] was considered to limit the number of simultaneous sound sources. This technique can be used to select objects falling on the path of a ray that originated at the user's position and viewing direction. However, empty areas within virtual objects (e.g., the hole in a donut-shaped object) or narrow surfaces could be overlooked by a single ray cast. To rectify this, we adopted a cylinder (a ray with volume) as our intersection tester (Figure 6). This allows us to check a set of objects within its radius, allowing larger tolerance for empty areas and reducing selection mistakes.

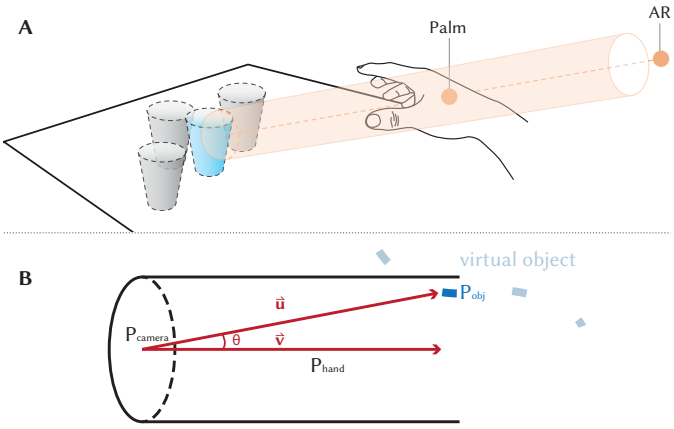

Figure 6. A) The cylindrical projection originated from the AR camera through the center of the user's palm determines which virtual objects emit sound. If multiple objects intersect with the cylinder, the closest one emits sound. B) A diagram illustrating an intersection condition between a point on the virtual object (P_{obj}) and the cylindrical projection. Light blue dots denote colliders on a virtual object.

The sound feedback is implemented in two steps. First, we construct a virtual cylinder passing through a users' AR camera and the center of the tracked hand. We define $\mathbf{v}$ as the vector pointing from P_{camera} to P_{hand}, and define $\mathbf{u}$ as the vector pointing from P_{camera} to any point P_{obj} on the collider of a virtual object, where $P_{obj} \in \mathbb{R}^3$. Then, we let $\theta = \arccos\left(\frac{<\mathbf{u},\mathbf{v}>}{\|\mathbf{u}\|\|\mathbf{v}\|}\right)$, where $<\mathbf{u},\mathbf{v}>$ denotes the inner product of $\mathbf{u}$ and $\mathbf{v}$. $\mathbf{1}_{[0,R)}$ denotes the indicator function of the interval $[0,R)$ where R is the radius of a reference cylinder. P_{obj} is within our cylinder when,

$$\mathbf{1}_{[0,R)}(\|\mathbf{u}\|\sin\theta)\|\mathbf{u}-\mathbf{v}\| \quad (2)$$

This returns the distance between P_{obj} and P_{hand} whenever P_{obj} is in the referred cylinder (Figure 6). $R = 4$ mm is for mid-sized virtual objects and to compensate for hand tracking error.

The second step is to modify the sound pitch V_{pitch} inversely based on the distance to the closest found object $D_{selected}$ and the maximum desired sound frequency F_{max}. The empirically chosen $s = 0.04$ is a scale factor for ARCore:

$$V_{pitch} = \frac{F_{max}}{1 + s \times D_{selected}}, \quad (3)$$

Haptics

Haptic feedback has been shown to be effective in confirming user intent in free-hand AR environments [7]. However, haptics cannot act on the manipulating hand in smartphone AR environments. As such, haptic retargeting [3] has been shown to be especially effective when visual perception dominates, and can also be useful without visual signals.

Given these findings, a 25 ms haptic feedback was implemented for two different situations: interaction state changes and virtual objects collisions. These signal to a user when the hand has touched the virtual object or when one object is placed within the collision boundaries of another. This parallels the physical world when, for example, placing a cup on a table induces vibrations in your hand as the cup makes contact with the table.

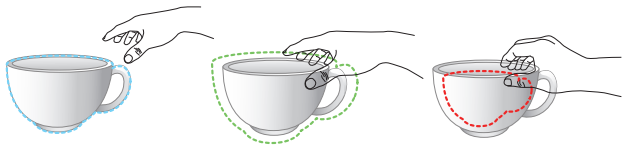

Figure 7. One accommodation dedicated to reduce failed grabbing due to a recessed grab. *Left:* Collision detection regions fit the volume of the virtual cup. *Middle:* This region expands to the center of the user's palm. *Right:* Once grabbed, the collider is set to 90% of the original collision detection range to ensure easy object release.

Accommodation for Manipulation

With the goal of creating a more reliable and intuitive free-hand experience when picking up, putting down, moving, and rotating virtual objects, we redesigned these fundamental mechanisms based on our findings about content manipulation (**I3**). First, we changed the collision detection structure to reduce errors and to make pick up and put down actions easier; then, we dynamically adjusted the collision detection range to account for different gestural behaviors and hand sizes. Finally, we used a manipulation and gesture stack to increase robustness. These accommodation mechanisms are described in more detail below.

Step 1: Accommodating Mixed Reality Physics

The redesigned collision detection structure used a two-collider scheme where one collider was dedicated to collision regions of virtual objects ($C_{physics}$), and the other collider with non-reactive behavior was dedicated to collision regions of the user's hand ($C_{interaction}$). Specifically, this gave us the freedom to change the collision regions, conditions, and sensitivities for interactions between the user's hand and virtual environments. For example, if two virtual balls collided, they could bounce off each other; at the same time, if the user's hand penetrated a ball, then the physics would be adjusted to avoid errors.

Step 2: Adaptive Collider Adjustments

Numerous cases in experience prototyping indicated failure in users' attempts to pick up and put down virtual objects due to lack of depth perception. To compensate for this, first we experimented with a fixed increase to the size of the virtual objects' colliders ($C_{interaction}$) to allow extra room for grabbing, similar to Grasp-Shell [34]. However, an effective incremental increase in collider size varies from one person to another, and is based on hand size and grasping angles.

To address this, an adaptive increment to the collider's size was considered for different hand sizes (Figure 7). We resize $C_{interaction}$ proportional to the longest finger of a user's hand [16]. Then, a three-stage grabbing procedure was engaged to ease interaction issues such as a recessed grabbing, where the closed hand is outside the collider's detection area. During this procedure, a collider starts at its *original* size, but increases until it reaches the center of user's palm when *at least two fingers* have reached the object. Once a grab is initiated, the collider's size *shrinks* for easier release.

In the implementation, C_{min} denotes the closest point from the virtual object O to the center of the user's palm P. We use a ray casting technique to find C_{min}, and scale the virtual object's

colliders from their center by λ such that C_{min} reaches the center of the user's palm:

$$\lambda = \frac{\|P-O\|}{\|O-C_{min}\|} \quad (4)$$

This technique can be applied to primitives, irregular shapes, and curved surfaces. One limitation is that when virtual objects are immediately adjacent to one another, the expansion of $C_{interaction}$ can cause one virtual object's collider to temporarily overlap with another virtual object's collider, resulting in unwanted or random selections. Using a stack to record the order and status of the last visited objects can mitigate this issue.

Step 3: Manipulation Stack

The stack stores virtual object IDs, whether an object has been touched by a user's hand or not, and which hand touched the object. This stack can hold up to n objects from the head of the stack, and allows users to select k objects from a single grab interaction. Since the last object reached by the hand is the first element in the stack, it becomes the selected item and this mitigates incorrect grabs when multiple object colliders overlap. Further, this stack helps to identify which hand the user is using and retains the flexibility to pick up multiple objects.

Accommodation for Behavior

Portal-ble contains an interactive calibration procedure to help users to establish a mental model for free-hand direct manipulation through observation and practice. This includes a series of training videos and scenarios eliciting the limitations of smartphone hand tracking. The system prompts the user to fully stretch their arms and to move as close to the hand sensor as possible until tracking is lost. During this process, the system records the minimum and maximum hand distance values, and uses these values to give visual feedback whenever a user's hand nears or reaches beyond the tracking limit.

Specifically, Portal-ble uses a red-zone visualization to warn about hand tracking limitations. The red-zone is only visible when the user's hand is near the minimum and maximum hand distance limit set in the calibration. When this happens, a red contour on the phone's screen gradually becomes visible. Once the manipulating hand is beyond the threshold, additional text appears instructing users to move their hands further away from or closer to the smartphone. This method helps users to find a balance between their preferred interaction zone and the range that the system is capable of detecting.

The bounds of the red-zone area are determined by two concentric spheres, with radii R_{min} and R_{max} from the earlier calibration. We check the distance d of the user's hand to the smartphone; a red-zone fades in when d is close to either the minimum or the maximum range, and it flickers when the hand is out of tracking. We determine these behaviors by changing the red-zone's opacity, where t decides the starting time:

$$Opacity = \begin{cases} 1-(d-R_{min})/t, & \text{if } R_{min} < d < R_{min}+t \\ (R_{max}-d)/t, & \text{if } R_{max}-t < d < R_{max} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

EVALUATION

By combining feedback and accommodation mechanisms, Portal-ble aims to improve users' awareness of current interaction state and tracking limitations and so improve the overall intuitiveness of smartphone-based free-hand interactions. To evaluate this, we tested Portal-ble for its ability to facilitate accurate perception and object manipulation with users who were free to move around with the system. This evaluation was undertaken to understand the overall efficacy of our system, its cognitive effect on users and its potential for everyday use.

Study Design

We compared Portal-ble to our initial smartphone AR prototype, which is used as a baseline with no feedback or accommodation mechanisms. For gesture prediction, both systems use the same SVM model with the same accuracy.

We test the research hypothesis: *Portal-ble will improve user depth perception and success in grabbing and manipulating virtual objects, and reduce user cognitive effort*. We designed two main tasks to evaluate our questions by measuring how fast participants completed tasks and how successful they were in grabbing virtual objects. Additionally, cognitive ratings were measured using the NASA-TLX assessment tool.

Task 1 (Approaching and Grabbing)

Motivation: This task explores how quickly and accurately participants navigate to and pick up distant virtual objects in a living space. Stationary objects are frequently tested in AR studies, but moving objects are particularly challenging because timing is critical: the user must quickly perceive and grab at the 3D location of the virtual object. Therefore, we create two subtasks: picking up a motionless object (Task 1.1) and picking up a moving object (Task 1.2). The moving object task is more difficult because it prevents the user from making relative adjustments between grab attempts.

Task Description: The first subtask for each participant is either Task 1.1 or Task 1.2, chosen at random. The participant begins at position x with a 60 $\times$ 60 cm physical table 3.5 m away. To prevent participants from using the physical spatial cues of the table, participants were instructed to look at the distant table surface only through the smartphone screen. After walking towards the table, the participant picks up a 10 $\times$ 10 $\times$ 20 cm virtual object three times. For Task 1.2, the object moves in a random direction at 20 cm/sec; when the object hits the table boundary, it changes to a new random direction. Once complete, this comprises three trials of one subtask.

Then, the participant returns to x and turns away from the table so that it is not visible. To minimize spatial learning effects and ordering effects, the experimenter moves the table in a random direction by a distance between 0 and 2.4 m during subtask switches (e.g., from Task 1.1 to Task 1.2), or every three trials. Then, the alternate subtask is performed.

Conditions: Picking up distant objects involves two perceptual issues: "walking towards the target" (**I1**) and "grabbing the target" (**I2**). Each issue requires a different accommodation to mitigate its challenges. To assess these accommodation mechanisms, participants were assigned to pairs of feedback

Table 1. Task conditions. Task 1 had two subtasks, Tasks 1.1 and 1.2. Each participant performed 6 trials for each condition, except for the physical condition which was only tested for a motionless object.

Task	Condition	Number of Trials
Task 1 (Grabbing)	Sound	3 (×2)
	Progress Wheel	3 (×2)
	Highlight	3 (×2)
	Haptic	3 (×2)
	Baseline	3 (×2)
	Physical	3
Task 2 (Manipulation)	Accommodations	3
	Baseline	3

conditions instead of testing one, three, or more conditions at once. These pairs are generated from our four feedback conditions: *sound* for **I1**, *progress wheel* for **I1**, *highlight* for **I2**, and *haptic* for **I2**, in addition to our *baseline* of no feedback. All four possible pairs were tested for the entire participant group with pre-generated and balanced orders and frequencies for these pairs. Additionally, participants completed a physical version of Task 1.1 with a stationary object, where they were asked to reach a physical object of the same shape and size as that of the virtual object. This was not done with a moving object (Task 1.2) due to the difficulty of moving the physical object in the same way as a virtual object. Table 1 details the conditions and number of trials for each task.

Measures: We collected completion time and success rate for each trial, and used a logging script to record the *timestamp*, *detected gesture*, *hand distance*, *body distance*, and *target object* of the participants' interactions. We also added an on-screen button to manually timestamp trial start and end times.

Task 2 (Manipulation)

Motivation: While users might easily grab an object, the finesse to orient and place it is a different operation. This task evaluates Portal-ble's designs that affect the user's ability to perform fine-tuned rotation, translation, and alignment of virtual objects where precise placement is important. Unlike Task 1, the goal of this task is to assess manipulation accuracy and quality rather than the speed of grabbing an object.

Task Description: Participants are asked to assemble a set of AR blocks into a virtual play house. Trials are completed by participants' satisfaction, giving up, or a three minutes timeout. They are given a printed photograph of a finished house which contains five virtual blocks: four rectangular and one trapezoidal. These blocks are initially generated with randomized orientation and location on the table.

Conditions: To account for ordering effects, participants began with either Portal-ble's accommodation mechanisms or the baseline setup. The starting order was pre-determined so that an equal number of participants started with each condition. For both conditions, each participant was expected to complete three different trials to build the same house.

Measures: We used a three-category rating scale to evaluate the quality of each AR house created by participants (see example houses representing each rating in Figure 8). Two of the authors rated the finished houses separately and the two ratings were averaged. The rating scale was defined as follows:

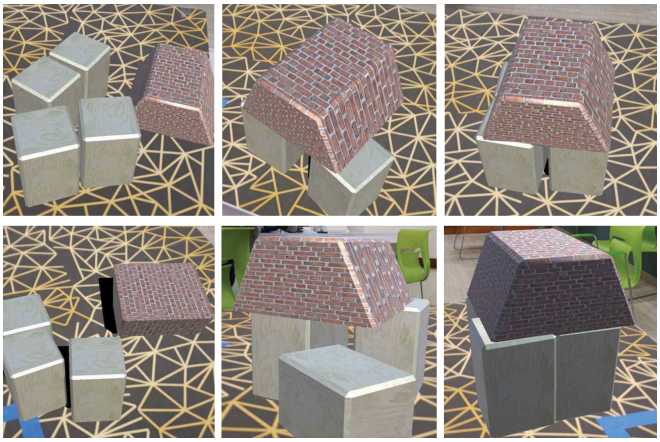

Figure 8. Examples of the scores assigned to houses constructed at varying levels of completeness for Task 2. Incomplete houses received 0 points, partially complete (having at least two pillars) houses received 0.5 points, and completed houses received 1 point.

- 0 if any component was missing, or the assembled house did not look like the photograph;
- 0.5 if the house was finished but some blocks were oriented incorrectly;
- 1 if the house looked like the photograph with correct orientation and no missing pieces.

Setup

The study took place in a large, furnished living room space to simulate everyday physical spaces (10×3 m), with a specific focus on walking up to and interacting with AR objects. Once participants arrived, they were given a tour of the space while experimenters explained the tasks. Before signing the consent form, participants were notified that the session would be recorded and were asked to think aloud. The experimenters then set up and configured Portal-ble for the participants, which included calibrating participants' hands and guiding them through Portal-ble's tutorial. The automatic and interactive tutorial walked participants through basic system functionalities, establishing a common understanding of "what a direct manipulation system is" and "what gestures can be recognized." The tutorial also showed participants the designed gestures for grabbing or releasing a virtual object (i.e., pinch and palm). After this introduction, participants were able to understand the basic concept of free-hand direct manipulation and any questions were addressed by the experimenters before the study began.

Participants

Electronic flyers were sent to students on various university listservs and advertised on social networking applications. Twelve participants (4 male and 8 female) were recruited for the study, ranging from 19–28 years old ($\bar{x} = 23$, $\sigma = 3$). Eight of the participants had prior experience with smartphone AR systems, but none had any experience with free-hand manipulation on smartphones. Participants were compensated $15 an hour for their time, with the actual average study taking 55 minutes.

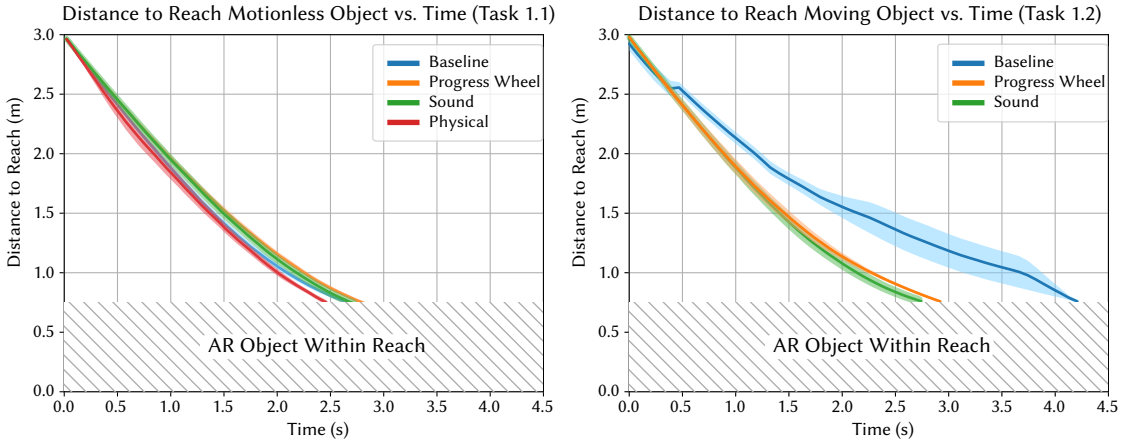

Figure 9. Mean distance between participants and virtual objects over time, in the grabbing task for motionless (Task 1.1, left) and moving (Task 1.2, right) virtual objects. The color-shaded area around each curve represents standard error. The physical object stays stationary during these trials and so only exists in Task 1.1. The graph shows that Portal-ble's near-range distance aid reduces the time to reach the moving virtual object in Task 1.2.

Procedure

At the end of every subtask or task, participants were asked to complete a NASA-TLX form, rating their effort, frustration, mental demand, performance, physical demand and temporal demand (perception of time). The NASA-TLX reports were collected using the official NASA app downloaded from the iOS App Store. The ratings were later combined into *Portal-ble* and *baseline* systems for comparison. At the end of the study, participants were asked to rate their preferences and rate the helpfulness of each condition on a 5-point Likert scale.

We collected a total of 464 trials (360 were for Task 1, 36 for the physical task, and 68 for Task 2), and logged the real-time distances traversed for each trial. One participant was only able to perform 2 trials for Task 2 due to time constraints. Another participant's distance data was lost due to a technical issue.

Results

For Task 1, we log-transformed the time-to-completion for grabbing virtual objects from all 12 participants, checked the normality (Mauchly's W, $p = 0.31$), and applied a one-way ANOVA with repeated measures to test the difference. Performance with Portal-ble was significantly faster than the baseline ($F(2,34) = 4.68$, $p = 0.016$). A post-hoc pair-wise comparison showed that Portal-ble's *haptic* feedback significantly improved the performance to Task 1.1 (Tukey HSD, $p = 0.032$) and *highlight* feedback significantly improved participant performance in moving objects in Task 1.2 (Tukey HSD, $p = 0.011$).

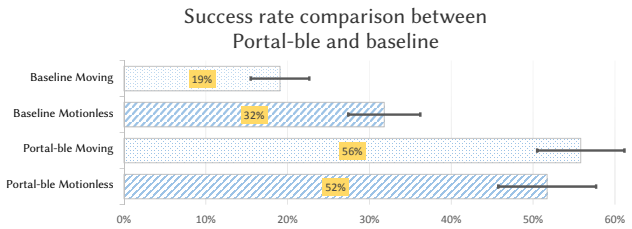

Figure 10. Participants using Portal-ble were more successful at grabbing virtual objects compared to the baseline, picking them up over half the time. In the baseline condition, the difference between the moving and motionless subtasks is large; for Portal-ble, the difference is small.

We normalized the real-time distance data with all participants and plotted the average distance over task time for each condition in the two tasks (Figure 9). The color-shaded area near each curve denotes standard error. Participants did not reach objects any faster in Task 1.1, but there was a substantial reduction in time to reach the virtual objects using Portal-ble's feedback compared to the baseline condition in Task 1.2.

The success rate is computed by dividing the number of successful grabbing interactions by the total number of attempted grabs. Each success rate was calculated per participant and compared group-wise. Our system had a significantly higher chance of allowing participants to grab both motionless ($t(11) = -2.75$, $p < 0.001$) and moving virtual objects ($t(11) = -4.16$, $p < 0.001$) over the baseline method (Figure 10). There was a summative evaluation for the quality of the manipulation as well. For Task 2, the authors rated a total build quality of 68 virtual houses, and found a significantly higher rating for houses assembled using Portal-ble ($t(33) = 2.51$, $p = 0.014$).

The NASA-TLX comparison showed that Portal-ble improved in every measure over the baseline method, with lower cognitive scores and higher perceived performance (Figure 11). We found that the overall cognitive score for

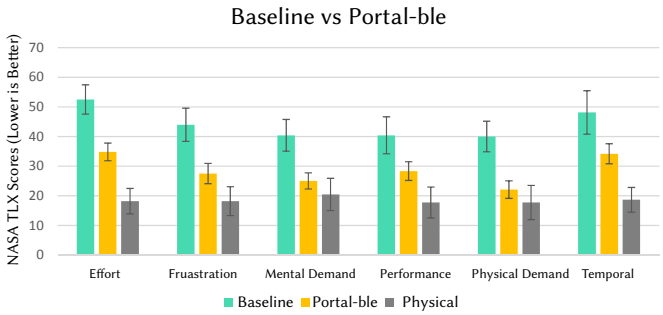

Figure 11. NASA Task Load Index assessment shows that Portal-ble had significantly less perceived effort, causes less frustration, has lower mental demand, higher perceived performance, and has less physical and temporal demand than the baseline. Manipulating physical objects naturally is still perceived as less demanding along all of these factors.

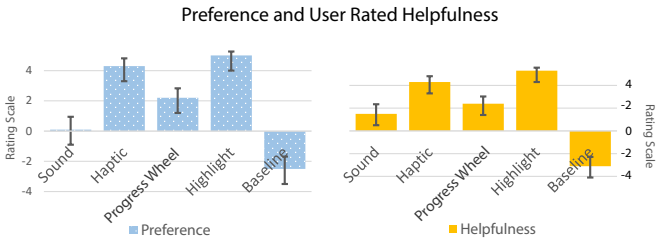

Figure 12. Users rated the manipulation feedback modes in Portal-ble to be more helpful than the baseline. Specifically, highlights and haptic feedback were particularly labeled as helpful.

NASA-TLX is significantly lower for Portal-ble compared to the baseline ($t(11) = 2.95, p = 0.014$). Further ratings collected from participants reflected significant differences among preferences ($F(4,50) = 3.64, p = 0.01$) and helpfulness ($F(4,50) = 4.90, p < 0.01$). Post hoc pairwise comparisons showed that Portal-ble's ratings in both preferences and helpfulness are significantly higher than the baseline.

Participants preferred close-range feedback such as the haptic and highlight feedback, which were rated as more helpful (Figure 12). Sound was less preferred, as five participants noted the sound was "annoying" and like an alarm, though they still rated it highly on helpfulness. P1 specifically mentioned that "the changing frequency helps me knowing the object is getting closer, but it also makes me nervous." These five participants all believed that the choice of specific sound used might have a different impact on their preferences.

DISCUSSION

The evaluation reveals an overall improvement in completion time, success rate, manipulation quality, and cognitive load. In addition, most users prefer Portal-ble to the baseline.

In Task 1, Portal-ble affected participant spatial perception differently at various distance intervals. In both Portal-ble and the baseline method, participants did not show discernible hesitation in approaching objects when they were more than 2.5 m away. However, estimating distances without feedback became increasingly difficult as participants approached moving virtual objects (Figure 9). The effect that occurs at the exact value of 2.5 m might be affected by the color or size of the virtual object. Future investigation could help to understand this effect.

In Task 2, there was a higher success rate in grabbing and lower cognitive load. The redesigned manipulation system was considered "joyful to use" by some participants. P1, P4, P5, P9, P11, and P12 specifically pointed out that the accommodation mechanisms were "much better [than the baseline]" when constructing the virtual house. Additionally, participants using Portal-ble were more confident in interacting with AR objects and in walking around the physical space.

Our results indicated that our feedback and accommodation mechanisms were more effective for difficult tasks: Task 1.2 and Task 2. One possibility is that participants are able to adapt to a lack of depth perception for stationary objects over time, but it takes longer for them to make that same adaptation for moving objects. In this respect, Portal-ble appears to reduce users' adaptation time to moving objects compared to the baseline condition. Further, in the baseline condition, some participants walked back and forth to locate the virtual object relative to the physical world when virtual objects were moving (Task 1.2), but no participants displayed these behaviors with Portal-ble. Finally, while Portal-ble allows participants to *approach* virtual objects with a similar performance to physical objects, we might not expect participants to be able to *grab* virtual objects as quickly as they can grab physical objects due to their lifelong experience with physical object manipulation.

Smartphone Free-hand Manipulation in the Near Future

How soon might smartphones be able to accurately estimate free-hand positions and gestures? Recent advancements in smaller GPU-accelerated CNN models for RGB cameras [24] suggest that this is sooner rather than later, as do improvements to HMDs with depth-camera-based hand tracking, such as the HTC Vive Pro and Hololens 2. With usability improvements and better tracking quality, free-hand tracking might become a standard input modality for smartphone AR interactions. Further, while we investigate the interaction space behind the smartphone with a world-facing camera, our study has yet to consider the hand interaction space created via a user-facing sensor. We leave this investigation for future work.

CONCLUSION

Based on the prior literature and design considerations, we iteratively constructed and improved a smartphone AR free-hand manipulation system to investigate user interface issues and provide feedback and accommodation mechanisms. First, an experience prototyping procedure was adhered to explore free-hand usability challenges. From observing users rearranging virtual objects and drawing in AR, we identified issues related to depth perception, manipulation, and user behaviors. Then, we addressed these issues by designing visual, audio and haptic feedback, manipulation accommodations, and user interface improvements. An empirical study showed that our final design is significantly more effective and satisfying, and less cognitively demanding for users as compared to a baseline implementation drawn from existing techniques from the literature.

We hope that exploring these usability issues will increase the popularity and research interest in smartphone-based free-hand manipulation. Portal-ble is a step towards exploring a portable augmented reality format where the smartphone is a portal into the virtual world, so we can interact with both the virtual and the physical in an intuitive way.

ACKNOWLEDGEMENTS

This work is supported by the National Science Foundation under Grant No. IIS-1552663 and by a gift from Pixar. We thank Fumeng Yang, Sara Gramley, Meng Kun, Andries van Dam, Leslie Welch, Neille-Ann Tan, Valerie Nguon, Nediyana Daskalova, Shaun Wallace, and Arielle Chapin for helping with editing and intellectual support. We also thank student researchers in the CSCI 2300 course at Brown University for conducting the experience prototyping sessions.

REFERENCES

[1] Leila Alem, Franco Tecchia, and Weidong Huang. 2011. HandsOnVideo: Towards a gesture based mobile AR system for remote collaboration. In *Recent trends of mobile collaborative augmented reality systems*. Springer, 135–148.

[2] John Aliprantis. 2019. Natural Interaction in Augmented Reality Context. *Visual Pattern Extraction and Recognition for Cultural Heritage Understanding* (2019), 50–61.

[3] Mahdi Azmandian, Mark Hancock, Hrvoje Benko, Eyal Ofek, and Andrew D Wilson. 2016. Haptic retargeting: Dynamic repurposing of passive haptics for enhanced virtual reality experiences. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 1968–1979.

[4] Myroslav Bachynskyi, Gregorio Palmas, Antti Oulasvirta, Jürgen Steimle, and Tino Weinkauf. 2015. Performance and ergonomics of touch surfaces: A comparative study using biomechanical simulation. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 1817–1826.

[5] Huidong Bai, Lei Gao, Jihad El-Sana, and Mark Billinghurst. 2013. Free-hand interaction for handheld augmented reality using an RGB-depth camera. In *SIGGRAPH Asia 2013 Symposium on Mobile Graphics and Interactive Applications*. ACM, 22.

[6] Huidong Bai, Gun A Lee, Mukundan Ramakrishnan, and Mark Billinghurst. 2014. 3D gesture interaction for handheld augmented reality. In *SIGGRAPH Asia 2014 Mobile Graphics and Interactive Applications*. ACM, 7.

[7] Carlos Bermejo and Pan Hui. 2017. A survey on haptic technologies for mobile augmented reality. *CoRR* abs/1709.00698 (2017). `http://arxiv.org/abs/1709.00698`

[8] Wolfgang Birkfellner, Michael Figl, Klaus Huber, Franz Watzinger, Felix Wanschitz, Johann Hummel, Rudolf Hanel, Wolfgang Greimel, Peter Homolka, Rolf Ewers, and others. 2002. A head-mounted operating binocular for augmented reality visualization in medicine-design and initial evaluation. *IEEE Transactions on Medical Imaging* 21, 8 (2002), 991–997.

[9] Marion Buchenau and Jane Fulton Suri. 2000. Experience prototyping. In *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*. ACM, 424–433.

[10] Volkert Buchmann, Stephen Violich, Mark Billinghurst, and Andy Cockburn. 2004. FingARtips: gesture based direct manipulation in Augmented Reality. In *Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*. ACM, 212–221.

[11] John G Buckley, Gurvinder K Panesar, Michael J MacLellan, Ian E Pacey, and Brendan T Barrett. 2010. Changes to control of adaptive gait in individuals with long-standing reduced stereoacuity. *Investigative ophthalmology & visual science* 51, 5 (2010), 2487–2495.

[12] Keith Cheverst, Nigel Davies, Keith Mitchell, Adrian Friday, and Christos Efstratiou. 2000. Developing a context-aware electronic tourist guide: some issues and experiences. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, 17–24.

[13] Sam Corbett-Davies, Andreas Dunser, Richard Green, and Adrian Clark. 2013. An advanced interaction framework for augmented reality based exposure treatment. In *2013 IEEE Virtual Reality (VR)*. IEEE, 19–22.

[14] Leap Motion Developer. 2017. Interaction Engine 1.1.0. (2017). `https://developer.leapmotion.com/releases/interaction-engine-110`

[15] Steven Feiner, Blair MacIntyre, Tobias Höllerer, and Anthony Webster. 1997. A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment. *Personal Technologies* 1, 4 (1997), 208–217.

[16] Ruggles George. 1930. Human finger types. *The Anatomical Record* 46, 2 (1930), 199–204.

[17] Leo Gombač, Klen Čopič Pucihar, Matjaž Kljun, Paul Coulton, and Jan Grbac. 2016. 3D Virtual tracing and depth perception problem on mobile AR. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 1849–1856.

[18] Zhenyi He and Xubo Yang. 2014. Hand-based interaction for object manipulation with augmented reality glasses. In *Proceedings of the 13th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*. ACM, 227–230.

[19] Iris Herbst, Anne-Kathrin Braun, Rod McCall, and Wolfgang Broll. 2008. TimeWarp: interactive time travel with a mobile mixed reality game. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*. ACM, 235–244.

[20] Otmar Hilliges, David Kim, Shahram Izadi, Malte Weiss, and Andrew Wilson. 2012. HoloDesk: Direct 3D Interactions with a Situated See-Through Display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2421–2430.

[21] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*. ACM, 1063–1072.

[22] Tobias Höllerer, Steven Feiner, Tachio Terauchi, Gus Rashid, and Drexel Hallaway. 1999. Exploring MARS: developing indoor and outdoor user interfaces to a mobile augmented reality system. *Computers & Graphics* 23, 6 (1999), 779–785.

[23] Wolfgang Hürst and Casper Van Wezel. 2013. Gesture-based interaction via finger tracking for mobile augmented reality. *Multimedia Tools and Applications* 62, 1 (2013), 233–258.

[24] Andrey Ignatov, Radu Timofte, William Chou, Ke Wang, Max Wu, Tim Hartley, and Luc Van Gool. 2018. Ai benchmark: Running deep neural networks on android smartphones. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 0–0.

[25] Varun Jain, Gaurav Garg, Ramakrishna Perla, and Ramya Hebbalaguppe. 2019. GestARLite: An On-Device Pointing Finger Based Gestural Interface for Smartphones and Video See-Through Head-Mounts. (2019).

[26] Minseok Kim and Jae Yeol Lee. 2016. Touch and hand gesture-based interactions for directly manipulating 3D virtual objects in mobile augmented reality. *Multimedia Tools and Applications* 75, 23 (2016), 16529–16550.

[27] Ernst Kruijff, J Edward Swan, and Steven Feiner. 2010. Perceptual issues in augmented reality revisited. In *2010 IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 3–12.

[28] Taehee Lee and Tobias Hollerer. 2007. Handy AR: Markerless inspection of augmented reality objects using fingertip tracking. In *Wearable Computers, 2007 11th IEEE International Symposium on*. IEEE, 83–90.

[29] Annette Mossel, Benjamin Venditti, and Hannes Kaufmann. 2013. 3DTouch and HOMER-S: intuitive manipulation techniques for one-handed handheld augmented reality. In *Proceedings of the Virtual Reality International Conference: Laval Virtual*. ACM, 12.

[30] Leap Motion. 2015. Leap Motion VR Best Practice Guidelines. (Jun 2015). `https://developer-archive.leapmotion.com/vr-best-practices`

[31] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. 2018. GANerated hands for real-time 3D hand tracking from monocular RGB. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 49–59.

[32] Franziska Mueller, Dushyant Mehta, Oleksandr Sotnychenko, Srinath Sridhar, Dan Casas, and Christian Theobalt. 2017. Real-time hand tracking under occlusion from an egocentric rgb-d sensor. In *Proceedings of the IEEE International Conference on Computer Vision*. 1284–1293.

[33] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. 2015. Hands deep in deep learning for hand pose estimation. *arXiv preprint arXiv:1502.06807* (2015).

[34] Thammathip Piumsomboon, David Altimira, Hyungon Kim, Adrian Clark, Gun Lee, and Mark Billinghurst. 2014. Grasp-Shell vs gesture-speech: A comparison of direct and indirect natural interaction techniques in augmented reality. In *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 73–82.

[35] Thammathip Piumsomboon, Adrian Clark, Mark Billinghurst, and Andy Cockburn. 2013. User-defined gestures for augmented reality. In *IFIP Conference on Human-Computer Interaction*. Springer, 282–299.

[36] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. 1996. The go-go interaction technique: non-linear mapping for direct manipulation in VR. In *ACM Symposium on User Interface Software and Technology*. 79–80.

[37] Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. 2014. Realtime and Robust Hand Tracking from Depth. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1106–1113.

[38] Gerhard Reitmayr and Tom Drummond. 2006. Going out: robust model-based tracking for outdoor augmented reality. In *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE Computer Society, 109–118.

[39] Javier Romero, Hedvig Kjellström, and Danica Kragic. 2009. Monocular real-time 3D articulated hand pose estimation. In *2009 9th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 87–92.

[40] Scott D Roth. 1982. Ray casting for modeling solids. *Computer graphics and image processing* 18, 2 (1982), 109–144.

[41] Hartmut Seichter, Jens Grubert, and Tobias Langlotz. 2013. Designing mobile augmented reality. In *Proceedings of the 15th international conference on Human-computer interaction with mobile devices and services*. ACM, 616–621.

[42] Byung-Kuk Seo, Junyeoung Choi, Jae-Hyek Han, Hanhoon Park, and Jong-Il Park. 2008. One-handed interaction with augmented virtual objects on mobile devices. In *Proceedings of The 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*. ACM, 8.

[43] Dong Woo Seo and Jae Yeol Lee. 2013. Direct hand touchable interactions in augmented reality environments for natural and intuitive user experiences. *Expert Systems with Applications* 40, 9 (2013), 3784–3793.

[44] Rajinder S Sodhi, Brett R Jones, David Forsyth, Brian P Bailey, and Giuliano Maciocci. 2013. BeThere: 3D mobile collaboration with spatial input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 179–188.

[45] Jaka Sodnik, Saso Tomazic, Raphael Grasset, Andreas Duenser, and Mark Billinghurst. 2006. Spatial sound localization in an augmented reality environment. In *Proceedings of the 18th Australia conference on computer-human interaction: design: activities, artefacts and environments*. ACM, 111–118.

[46] Jie Song, Gábor Sörös, Fabrizio Pece, Sean Ryan Fanello, Shahram Izadi, Cem Keskin, and Otmar Hilliges. 2014. In-air gestures around unmodified mobile devices. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, 319–329.

[47] N. Sugano, H. Kato, and K. Tachibana. 2003. The effects of shadow representation of virtual objects in augmented reality. In *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings.* 76–83.

[48] Bruce Thomas, Benjamin Close, John Donoghue, John Squires, Phillip De Bondi, Michael Morris, and Wayne Piekarski. 2000. ARQuake: An outdoor/indoor augmented reality first person application. In *Wearable computers, the fourth international symposium on*. IEEE, 139–146.

[49] Bruce Thomas, Victor Demczuk, Wayne Piekarski, David Hepworth, and Bernard Gunther. 1998. A wearable computer system with augmented reality to support terrestrial navigation. In *Wearable Computers, 1998. Digest of Papers. Second International Symposium on*. IEEE, 168–171.

[50] Matt Whitlock, Ethan Harnner, Jed R Brubaker, Shaun Kane, and Danielle Albers Szafir. 2018. Interacting with Distant Objects in Augmented Reality. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 41–48.

[51] Hui-Shyong Yeo, Byung-Gook Lee, and Hyotaek Lim. 2015. Hand tracking and gesture recognition system for human-computer interaction using low-cost hardware. *Multimedia Tools and Applications* 74, 8 (2015), 2687–2715.

[52] Shahrouz Yousefi, Mhretab Kidane, Yeray Delgado, Julio Chana, and Nico Reski. 2016. 3D gesture-based interaction for immersive experience in mobile VR. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*. IEEE, 2121–2126.