Contents lists available at ScienceDirect

International Journal of Human - Computer Studies

journal homepage: www.elsevier.com/locate/ijhcs

An examination of mobile phone pointing in surface mapped spatial augmented reality

Jeremy Hartmann *, Daniel Vogel

University of Waterloo, Waterloo, Canada

ARTICLE INFO

Keywords:
Augmented reality
Mobile input
Pointing
Fitts' modeling

ABSTRACT

We investigate mobile phone pointing in Spatial Augmented Reality (SAR), where digital content is mapped onto the surfaces of a real physical environment. Three pointing techniques are compared: raycast, viewport, and direct. A first experiment examines these techniques in a realistic five-projector SAR environment with representative targets distributed across different surfaces. Participants were permitted free movement, so variations in target occlusion and target view angle occurred naturally. A second experiment validates and further generalizes findings by strictly controlling target occlusion and view angle in a simulated SAR pointing task using an AR HMD. Overall, results show raycast is fastest for non-occluded targets, direct is most accurate, and fastest for occluded targets in close proximity, and viewport falls in between. Using the experiment data, we formulate and evaluate a new Fitts' model combining two spatial configurations in a SAR pointing task to capture key characteristics, initial target occlusion, target view angle, and user movement. Analysis shows it is a better predictor than previous models.

1. Introduction

Spatial Augmented Reality (SAR) (Raskar et al., 1998; 1999) places digital content directly into a real physical environment. One application of SAR is to create immersive environments that differ significantly from physical reality (Jones et al., 2013), often for gaming or virtual teleportation (Pejsa et al., 2016). This typically involves covering and hiding large portions of real surfaces and objects with textures, often creating illusions of virtual 3D objects (Dolce et al., 2012; Hartmann et al., 2019; Jones et al., 2014; Oswald et al., 2014). In contrast, SAR can be applied in a more integrated and subtle way, where real surfaces and objects are selectively augmented with 2D digital information. We refer to this as "surface mapped" SAR, since it relates to SAR surface shading (Raskar et al., 2001). Essentially, every surface becomes a potential display, but without the restrictions from AR glasses, since the user very consciously exists in the real environment.

Such an environment could be used to facilitate cross-device interaction, for example content from a mobile phonecan be spread into underutilized spaces for the purpose of awareness (like weather conditions), notifications (like upcoming meetings), visualizations (like maps), or sharing content (like photos). Techniques already exist to track the 6-DOF position of a phone (Mur-Artal and Tardos, 2016; Ondruska et al., 2015) and to detect when it touches a surface (Hardy and Rukzio, 2008; Lopes et al., 2011; Schmidt et al., 2012). Enabled by this, the phone could be a ubiquitous input device for surface mappedSAR.

Mobile phonepointing has been explored with large displays (Seifert et al., 2013a), multi-display environments (Bragdon et al., 2011), hand-held projectors (Molyneaux et al., 2012), and using "viewport AR" in relatively planar scenes from a fixed perspective (Boring et al., 2009; Rohs and Oulasvirta, 2008; Rohs et al., 2011). In general, mid-air device pointing in AR and VR has assumed immersive or floating 3D targets (Benko et al., 2014; Teather and Stuerzlinger, 2010), while work examining surface mappedSAR has kept the user at a fixed location in a small desktop setting (Gervais et al., 2015), or in an essentially empty room (Petford et al., 2018).

We compare three popular mobile phonepointing techniques in surface mappedSAR. The techniques are adapted from other contexts:

* Corresponding author.
E-mail addresses: j3hartma@uwaterloo.ca (J. Hartmann), dvogel@uwaterloo.ca (D. Vogel).

https://doi.org/10.1016/j.ijhcs.2021.102662
Received 10 January 2020; Received in revised form 23 April 2021; Accepted 25 April 2021
Available online 6 May 2021
1071-5819/© 2021 Elsevier Ltd. All rights reserved.

raycasting from large displays, *viewport* selection from mobile AR, and *direct* contact of the phone from tabletops (Fig. 1). Our work significantly extends the initial analysis and results presented in Hartmann and Vogel's late-breaking-work poster (2018)[1]. In our first experiment, we evaluate mobile phonepointing in a realistic projection-based SAR environment (Fig. 1). The results identify key characteristics that influence pointing performance: the degree of target occlusion due to environment geometry, the target view angle relative to the user, and the amount of user movement required. Our second experiment tests these key factors in a highly controlled simulation of SAR pointing tasks using a stereo AR head-mounted display. Using data from both experiments, we develop a predictive model for mobile phonepointing in SAR that outperforms previous pointing models and further shows how the key characteristics represent the task difficulty. The experiment data and code is available[2].

In summary, our work makes two contributions: (1) empirical evidence for the relative performance of mobile phone pointing in SAR, showing that *raycast* is fastest for non-occluded targets, *direct* is most accurate, and fastest for occluded targets in close proximity, and *viewport* falls in between; (2) a new predictive model for surface mapped SAR pointing that incorporates three key characteristics, target occlusion, the user's spatial movement, and target viewing angle, by modelling the initial and ending relationships between the user and target.

2. Related work

Our work relates to previous evaluations of mobile phonepointing, including evaluations using similar mid-air hand-held devices like laser pointers. For the most part, these have been conducted in environments other than SAR, specifically large displays, multi-display environments, tabletops, and viewport AR. Although some VR and 3D user interface pointing studies have investigated hand-held device pointing, they focus on 3D virtual targets, not 2D targets fixed to planes of differing orientations and positions like our surface mappedSAR environment.

2.1. VR and 3D user interface pointing

There is extensive research in VR target selection (Boritz and Booth, 1997; Dang, 2007). Early work by Bowman and Hodges (1997) and Bowman et al. (1999) compared 3D pointing techniques like raycasting, world-in-miniature, and Poupyrev et al.'s (1996) go-go distant reaching method. Cashion et al. (2012, 2013) evaluated selection techniques in five game-like virtual environments with varying degrees of object density and dynamics. Teather and Stuerzlinger studied effects of stereo displays and passive haptics on target selection, and found objects viewed above the display degrade performance, but passive haptics improves throughput (Teather and Stuerzlinger, 2010; 2011). They later explored the effects of target depth on selection for both raycast and mouse-based input for a small stereoscopic display (Teather and Stuerzlinger, 2013). Results show that mouse-based input performed best overall.

There have been multiple prior extensions of Fitts' law to 3D tasks, Murata and Iwase (2001) examined hand pointing using an electromagnetic tracking system for a finger, and created a revised model that included movement angle direction. Grossman and Balakrishnan (2004) created a trivariate model tested on a 3D volumetric display based on Accot and Zhai's (2003) weighted bivariate with added direction and depth.

Pointing in a surface mappedSAR is different than 3D immersive pointing. In this type of SAR, there is no illusion of 3-D objects, all targets are placed on geometry in the physical world. As such, they will conform to the mostly planar surface it is displayed on. This means Fitts' models designed for 3D, like the trivariate model or Murata and Iwase's model, are incompatible since surface mapped 2D targets do not have depth or ordered direction vectors. Our evaluation and predictive model only consider targets that are strictly adherent to targets on geometric surfaces.

2.2. Large display and tabletop pointing

Early work by Myers et al. (2001) used a laser equipped Personal Digital Assistant (PDA) in a large display pointing technique that combines raycasting with fine tuning the selection on the PDA screen. Expanding on this, PointerPhone (Seifert et al., 2013a) report on a qualitative study using a laser equipped mobile phoneto raycast point at a large display. Bergé et al. (2014) and Nancel et al. (2015) examine related raycast variations. TractorBeam (Parker et al., 2005) is a pen-based technique to select distant targets on a tabletop display using

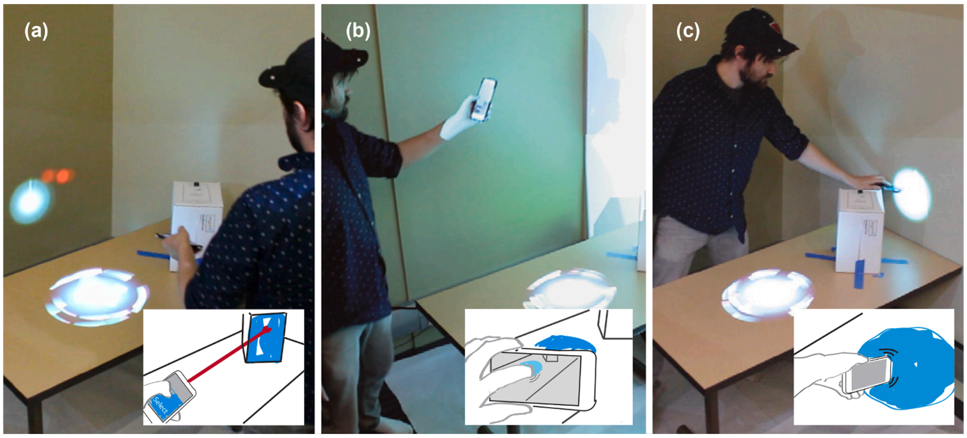

Fig. 1. Mobile phonepointing in the surface mappedSAR environment used in Experiment 1: (a) the raycast technique uses an invisible ray emanating from the phone to point at the desired target, with a tap on the phone screen to select it; (b) the viewport technique views targets through a simulated rear camera and selection is by tapping on the target on the touch screen; and (c) the direct pointing technique uses the phone itself to directly touch a target.

[1] This 5-page extended abstract describes Experiment 1 with basic analysis for movement time and a two-level form of target occlusion. The present paper provides a new refined analysis of Experiment 1, which includes an examination of occlusion at multiple levels, adds the important aspect of target view angle, and reports on an expanded set of metrics, like target error and user movement.

[2] https://git.uwaterloo.ca/exii-group-hci/sar-pointing-analysis.

raycasting. It demonstrates and evaluates how raycasting can become a form of direct interaction when the target is close enough to touch using a zero-length ray.

A popular selection technique uses the view of a mobile phone camera to interact with distant objects, called viewport pointing. Baldauf et al. (2012) proposed a marker-less tracking method for this style of viewport interaction. Machuca et al. (2014) explored viewport pointing with public displays and real objects. Both projects used a touch on the viewport image, neither report a controlled study.

Boring et al. (2010) TouchProjector phone viewport technique uses custom AR-like tracking to localize and then select content on a large display. Since their focus was on distant targets, they also explored enhancements to basic viewport pointing such as zooming and freezing the image. We limit our exploration to standard non-zoom viewport pointing since it is simpler and commonly used in current AR phone applications.

Investigating raycast performance on large displays, Jota et al. (2010) compared four 6DOF wand techniques while Kopper et al. (2010) evaluated four pointing models. Both proposed modified Fitts' law models that incorporate the angular width and distance from the user's perspective. We discuss these models in detail when we derive our own model in a later section. Mobile phoneshave also been used to point at targets on digital tabletops. Schmidt and colleagues explore a range of direct manipulation styles that fuse three devices: mobile phones, tabletops, and styluses (Schmidt et al., 2010; 2012). They use this to expand the interaction space for cross-device interaction.

2.3. Multi-display environment pointing

Mobile phonesand similar devices have been used for pointing in Multi-display Environments (MDEs). For example, Rekimoto investigated pen interaction as a means to transfer data between different computers (Rekimoto, 1997), tapping a pen between a large wall display, desktop display, or palmtop display is a form of direct input pointing. Bragdon et al. (2011) explored hybrid MDE interactions using mobile phoneraycast pointing, touch, and hand gestures that utilize multiple large displays and private displays like tablets. Nacenta et al. (2005) evaluated six pointing techniques in a simple multi-display environment that consists of a single large table and tablet computer. They later proposed a taxonomy for cross-display interaction based on the techniques properties, like spatial reference, configuration, and control (Nacenta et al., 2009). Their perspective cursor technique was used to control a mouse across multiple displays, dynamically mapping them based on perspective and spatial relationship of the user (Nacenta et al., 2006). Phone-like wands have also been explored, like XWand (Wilson and Shafer, 2003) and GyroWand (Hincapié-Ramos et al., 2015) that use a form of raycasting for pointing at interactive content in the world.

Proxemic interaction is an extension of MDEs, where distances between devices, people, and objects set the context for input. Marquardt et al. (2011) explored proxemic relationships using mobile phoneraycast pointing in some demonstrations. She et al. (2013) utilize the relative orientation between a mobile phoneand multiple large displays to signal a selection; essentially a raycasting heuristic. They demonstrated their technique in a 7-display environment surrounding multiple users.

2.4. SAR and AR pointing

Rohs and Oulasvirta (2008; 2011) evaluated "magic lens mobile phonepointing" at near-planar scenes, like distant buildings. Pointing is done through the camera-view of the phone, where the phone is first positioned in physical space near the target, then fine-tuned in virtual space in the phone display. We refer to this general type of interaction as viewport pointing. They proposed and tested a two-part model based on Fitts' law, that splits physical and virtual pointing phases into two terms. This modelling approach inspired our two-part model. Note, Rohs and Oulasvirta use a fixed, central viewport selection point, not a secondary tap on the screen like Boring et al's TouchProjector and the viewport technique we test in our experiments.

Gervais et al. (2015) evaluated pointing in a small desktop SAR environment while seated and stationary. A mouse and tablet were used, and the environment included targets on different faces of objects. They report standard Fitts' law holds when selecting targets on abnormal geometry. MeetAlive (Fender et al., 2017) used mouse pointing in a SAR environment to facilitate meeting productivity, which was largely contained to four flat walls and large boardroom table. Similarly, Petford et al. (2018) compared mouse and raycast pointing in a similar SAR environment that was constrained to four flat walls and a ceiling. They found the mouse to be fastest for targets in front of the user and raycast for targets behind. *Mano-a-Mano* (Benko et al., 2014) examined selection in a large room-sized SAR environment using a wand for mid-air selection. In contrast, Molyneaux et al. (2012) demonstrate direct touch and indirect shadow-based interaction techniques in a projector-based SAR system.

Pointing studies in AR and VR have focused almost exclusively on immersive 3-D object pointing (Benko et al., 2014; Hincapié-Ramos et al., 2015; Teather and Stuerzlinger, 2011) or AR pointing at near-planar scenes (Boring et al., 2009; Rohs and Oulasvirta, 2008; Rohs et al., 2011). Raycast, viewport, and direct mobile phonepointing techniques have been used with large displays, MDEs, and AR, but to our knowledge, never compared directly in SAR. In fact, few pointing studies have been conducted in SAR at all. Gervais et al. (2015) used a small desktop SAR environment, limited target variations, and unique interaction techniques controlled by a conventional mouse or tablet. Benko et al. (2014) conducted hand-pointing tasks within SAR, but this is in context of a view dependent rendering of 3-D objects. In contrast, we investigate popular mobile phonepointing techniques in a larger and more complex SAR environment.

3. Mobile phone pointing in SAR

We briefly describe our surface mappedSAR technical infrastructure, then provide details for the three mobile phonepointing techniques to be compared.

3.1. SAR system and environment

The setup occupies a corner of a room, occupying approximately 4 × 4 meters of floor space (Figs. 1 and 2). Mounted in the ceiling are 5 digital projectors, 6 Microsoft Kinect cameras (each connected to an IntelNUC Core i7-7567U), and a 10-camera Vicon (Vera/Bonita) tracking system. Tracker 3.5.0 software on a dedicated server tracks the 6DOF position of a mobile phoneand a person's head. The phone tracking object is a custom-printed phone case with seven 6.4mm spherical markers, and the head tracking object is a ball cap with five markers attached to the brim.

The main server (Windows 10, Core i7-6850K) is connected to the Vicon server and IntelNUCs using a 10Gb LAN network. Projectors and Kinects are calibrated using the RoomAlive toolkit (Jones et al., 2014), with the 3D room reconstruction imported into Unity3D. Manual adjustments to geometry position, and design tricks like texture blending and transparency, compensate for limited precision of projector alignment and room reconstruction. A Unity3D 5.6 application processes tracked objects, enables two-way mobile phonecommunication, and renders projection-mapped content with all projectors at 60 FPS using twin GTX 1080 GPUs.

The mobile phoneis a Google Pixel 2 running Android Nougat 7.1 (5.0" display, 149 × 74 × 11mm including case). A custom app enables the server to render a simple interface for experiment control and communicates status such as current motion tracking confidence.

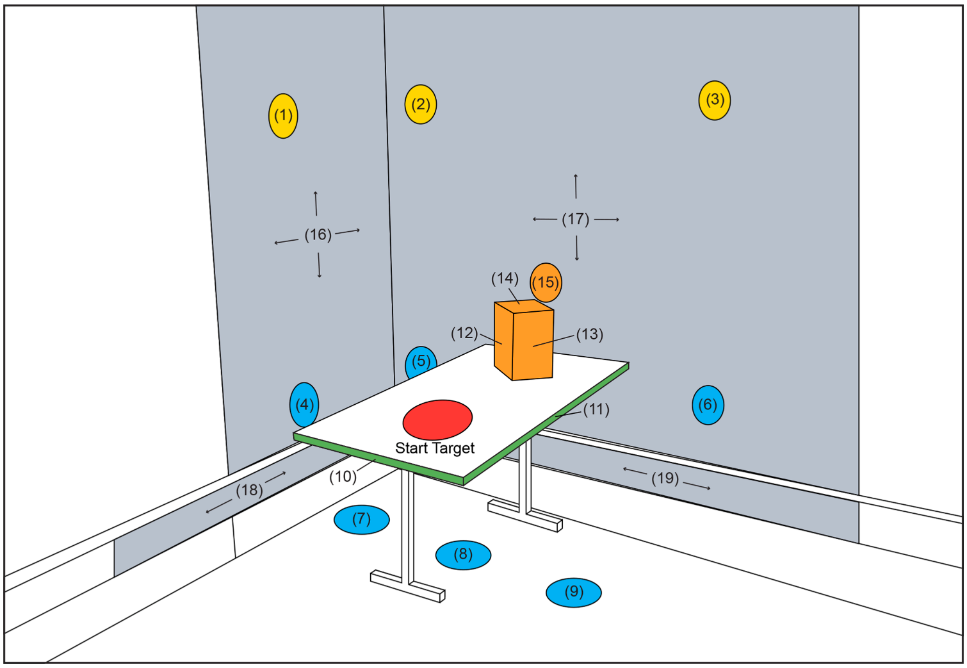

Fig. 2. Illustration of the SAR experiment environment showing target sizes and locations. For this figure, targets coloured yellow are in the HIGH group, gray are in the LARGE group, orange are MID group, green are TABLE group, and light-blue are in the LOW group. The start target is the red circle on the table, and the user's start position is located 3 m away from each corner wall, placing them directly opposite of where the walls intersect.

3.2. Mobile phone pointing techniques

Using the system above, we created three mobile phonepointing techniques suitable for SAR.

3.2.1. Raycast pointing

Previous mobile phoneraycasting techniques used a laser (Myers et al., 2001; Seifert et al., 2013b), or a geometric ray based on 3D tracked position (Jota et al., 2010) as we do. To use the technique, the user holds the mobile phonewith either their right or left hand, points the front end at a target, and taps the screen with their thumb to select (Fig. 1a). Since we accurately track the mobile phone's 3D position, and we have a 3D scan of the environment geometry aligned with the real world, we use a virtual ray to test intersections with virtual surfaces and objects. At the point of intersection, a red cursor is displayed on the surface.

3.2.2. Viewport pointing

Using the phone's camera like a viewport to select content is a common approach for virtual content selection. Implementations, like in Rohs and Oulasvirta (2008) and Rohs et al. (2011) work, use a single fixed cross-hair at the centre of the screen. Instead, we chose a more versatile method where targets are selected anywhere in the viewport by tapping directly on the screen (Boring et al., 2009). To use the technique, the user roughly frames the desired target using the phone like a camera, then taps the desired target in the display (Fig. 1b).

Typically, a live camera feed with computer vision tracking is used for viewport techniques. However, mapping a touch to a precise physical world location using a live camera view is challenging, and can be unstable and hard to accurately control using current AR methods. To avoid these potential confounds, our system uses a 3D rendering of the camera view synchronized with the SAR server. A virtual camera is matched to the position and orientation of the physical phone's rear camera, which is tracked by the Vicon system and able to achieve a higher degree of accuracy than current on-phone methods. By configuring the virtual camera to use the same 60° field-of-view as the real phone camera, and the 3D scanned and calibrated room geometry creating a one-to-one mapping between physical and virtual worlds, an accurate rendered camera view can be produced.

Boring et al. (2010) enhanced a standard viewport technique with several variations of zoom control (combined with selective frame freezing), tuned for selecting targets on distant displays with direct touch from a finger. Using direct touch on the display has the advantage of not requiring the user to precisely aim the phone, making the action more different than the precise phone aiming action required with raycasting. Tapping on small targets in the phone display does introduce a fat finger problem (Siek et al., 2005), but the user is free to move closer to the target to increase its overall size in the display. This worked well for our studies, but if the target is very small or the user is unable to move closer, then enhanced viewport techniques like Boring et al. propose can be used. One reason were not using zoom for enhanced interactions in our study is to keep the viewport technique under examination elemental, robust, and simple to use. Notably, most current viewport-style AR mobile phoneapplications using Apple's ARKit or Android ARCore also do not use zoom, so our viewport implementation is ecologically valid.

3.2.3. Direct pointing

Direct mobile phoneinput has primarily been used in the context of tabletops (Schmidt et al., 2012) and large displays (Hardy and Rukzio, 2008), where the mobile phoneacts as an extension of a person's hand. To use the technique, the user holds the mobile phonewith either their left or right hand and physically taps the currently active target with a corner, side, or face (Fig. 1c). Contact of the mobile phoneto surfaces and objects is triggered when a bounding box constructed around the phone intersects with the scanned 3D geometry of the environment.

3.2.4. Justification for technique selection

These three techniques were selected for four reasons. First, they represent relatively common methods for SAR and related contexts, as shown by our survey of related work. Second, all use a form of absolute pointing. Implementing a relative method, like Nancel et al. (2015) method for large displays requires a relative cursor which is challenging for multi-surface complex geometry in surface mappedSAR. Third, they are simple and can each be considered an elemental part of a more advanced implementation or a hybrid combination. For example, advanced pointing techniques, like Go-Go (Poupyrev et al., 1996) and Semantic Snarfing (Myers et al., 2001), are both built on the basic raycasting technique. Finally, our three techniques cover a range of pointing technique paradigms suitable to SAR. For example, they span at-distance and direct contact styles, both of which have been demonstrated with other input devices in a SAR context in systems like RoomAlive (Jones et al., 2014).

4. Experiment 1: ad-hoc SAR setting

This within-subjects experiment compares the three mobile phone-pointing techniques in a realistic ad-hoc surface mappedSAR environment. For each technique, the participant is free to move around the space during the selection task of the 2D targets. Instead of strictly controlling target size and width in the traditional sense, we created a constrained, but representative environment geometry, and selected target sizes and positions to represent content that might exist in a future where SAR is ubiquitous. A start target controls the user's initial position, but they can move freely after to select the required target. With the same task conditions across techniques, we examine how different segmentations of targets, such as by size and position, by initial target occlusion, or by target view angle, affect key task performance metrics like time, error, and user movement. The results lead to the identification of key characteristics, including target occlusion and target view angle, that are investigated in a controlled setting in Experiment 2. We also use the data from this experiment to develop and test a predictive model of SAR pointing based on these same key characteristics, explained later in this paper.

4.1. Participants

We recruited 18 participants, ages 21–50, 13 male, 5 female, 1 left-handed. All used a mobile phone every day. Remuneration was $10.

4.2. Apparatus

The SAR system described above is used in an environment containing a table with a small box on top (Fig. 2. The 61 × 59 × 122 cm table is positioned orthogonal to one wall and sits approximately 56cm from a parallel wall. On top of the table sits a 17 × 27 × 20 cm cardboard box rotated approximately 30° and sits 20cm from the wall. A large portion of the floor, the two corner walls, the table, and box were all covered by the light produced by the projectors, which were orientated to minimize shadows and maximize coverage over all surfaces in the environment. The system was calibrated within a 1 cm tolerance and all input tracking and target hit detection for measured trials used the Vicon, which is accurate to one mm or less. If tracking was lost during a trial, the phone notified the participant by vibrating and turning the screen red. However, during the experiment, tracking was rarely lost.

Our surface mappedSAR setup can be thought of as a Multi-Display Environment (MDE) with approximately 11 "displays", which are different surfaces in the room with very different sizes, orientations, and positions (with some hidden). Fig. 2 illustrates these surfaces. There are 2 large walls (large grey areas in Figure), 2 surfaces along different baseboard mouldings (rectangular grey area below wall surfaces), 1 floor surface, 1 table top surface, 2 table edge surfaces (thin green rectangles), and 3 surfaces forming the two sides and top of the box (shown in orange). In the experiment, we use most surfaces as one type of target (with the exception of floor and table top) and for large surfaces, we also display smaller circular targets mapped into a surface. These are explained below. Similar to the work by Molyneaux et al. (2012), we restricted our experiment to a room-sized environment as this more closely replicates a scenario where SAR would be used.

4.2.1. Logging and metrics

During trials, the system logged when each target was selected and whether selection errors occured to calculate the primary dependent variables of *Movement Time* and *Error Rate*. In addition, other data was logged: the position and orientation of the participant's head, the phone, and each target; all touchscreen input; and technique events and states (such as the 3D raycast cursor position). These are used to calculate a dependent variable for *Head Movement*, and a metric called the *visibility ratio* that determines how much of the target is occluded from the participant's perspective.

The *visibility ratio* is determined as follows. The system uses Unity to render 224 × 224px views of the full 3D scene from two virtual cameras (created in the same Unity scene) attached to the participant's head, one matching head orientation as a proxy for gaze, and the other oriented to the next target to be selected. For each camera, there are two rendering passes: one only containing the target, and the other containing the target with all scene objects that may occlude it. The proportion of the target in the second render relative to the first is the target's *visibility ratio* for each virtual camera.

4.3. Task

We imagine an environment where users interact with content on any surface. Consider a SAR office: pointing at a wall could place a large visualization like a map, pointing at the edge of a desk could silence a notification displayed there (Joshi and Vogel, 2019), pointing at the floor could open an application for viewing photos, and so on.

The experiment task was to select two targets in sequence as quickly and accurately as possible. Targets were rendered on real surfaces using our SAR system. The targets were bright and easy to locate. Auditory feedback was given for successful and unsuccessful target selections. The participant had to successfully select the target to complete the trial, but all trials with one or more errors are noted in the log. The first target was a circular *start target* (r = 18cm). The centre of the target was placed at a 30cm offset from the edge of the table. The second measurement target could be either a *circle* (r = 13cm) or a *rectangle* of varying dimensions. There were 19 targets grouped into five types, HIGH, MID, LOW, TABLE, and LARGE. Target positions, shapes, and sizes are illustrated in Fig. 2 and explained below:

HIGH — Composed of three circular targets positioned slightly above an average person's gaze (∼ 176 cm) See Fig. 2, yellow targets 1, 2, and 3.

MID — Composed of a circular target placed on the wall behind the box, and three rectangular targets mapped to the three sides of the 17 × 27 × 20 cm box. See Fig. 2, orange targets 12, 13, 14, and 15.

LOW — Composed of six circular targets placed on the floor or on the wall 20 cm below the table height. See Fig. 2, light-blue targets 4, 5, 6, 7, 8, and 9.

TABLE — Composed of two long thin rectangular targets placed along the front and side edges of the table each approximately 3 cm high and between 50 and 100 cm wide). See Fig. 2, green targets 10 and 11.

LARGE — Composed of four rectangular targets: two large rectangles covering the entire wall each approximately 300 cm by 300 cm and two rectangles conforming to the shape of two baseboards each approximately 25 cm by 100 cm, the bottom 30 cm above the floor. See Fig. 2, grey walls and baseboards 16, 17, 18, and 19.

Each target was chosen to replicate realistic scenarios that may be encountered in future SAR environments. The motivation for rectangular targets was to analyze pointing on full faces of geometry (like walls and edges). We give example applications above. The circular targets represent specific content locations. In SAR, the dimensions of targets is complicated by the user position and other geometry, but the model we develop later accounts for actual target size as it appears in the environment by considering view angle and occlusion.

Unlike classic Fitts' studies (Gervais et al., 2015; Teather and Stuerzlinger, 2011), we do not use a variation of the ISO/TS 9241-411

(0000) task. The ISO standard uses a radial set of circles around a centre point, but given the amount of geometric variation within SAR, any attempt to enforce a controlled circular pattern mapped onto the environment would render the control of distance and size nearly impossible. Considering that SAR is conforming to the physical environment, we designed this initial study to investigate pointing at targets representing possible real-world content placement. In a second study that follows, we use AR to simulate key SAR pointing task configurations with strict controls on target width, location, and size.

4.4. Design and protocol

The design is fully within-subjects. The primary independent variables are TECHNIQUE (3 levels: VIEWPORT, RAYCAST, and DIRECT) and TARGET (19 different targets spanning five categories: HIGH, MID, LOW, TABLE, and LARGE). The ordering of TECHNIQUE for each participant was counter balanced using a balanced Latin square. For each TECHNIQUE, the participant completed 5 BLOCKS of 19 TARGET selection tasks presented in random order. Recall that each target selection begins with a fixed start target, so each task sequence from start target to measurement target is a measurement trial.

Before the start of the experiment, each participant was given brief instructions on how to use each of the techniques, and told to be as fast and as accurate as possible. Participants were free to move around the space, but were required to return to the starting position at the beginning of each block. No other instructions were given. For each technique, a short practice session preceded the five blocks of measured trials. Each participant completed a short post-experiment questionnaire rating each technique on four subjective measures using a 1–10 scale: ease-of-learning, comfort, ease-of-use, and overall performance. The entire session lasted approximately 30 minutes.

In summary: 3 TECHNIQUES × 5 BLOCKS × 19 TARGETS, resulting in 285 data points per participant.

4.5. Results

Repeated measures ANOVA and posthoc t-tests with Holm correction were used for all measures. When sphericity was violated, degrees of freedom corrected using Greenhouse-Geisser ($\epsilon < 0.75$) or Huynh-Feldt ($\epsilon \geq 0.75$). Time data was aggregated using the median to account for a skewed distribution, and a BoxCox transformation (Box and Cox, 1964) corrected non-normal time data when necessary. 78 outliers more than 3 standard deviations from the mean target time were removed (1.5%).

4.5.1. Learning effect

We are interested in practised performance, so we verified there were no large differences in task times across subsequent blocks. There was no effect of BLOCK on *Movement Time* for RAYCAST ($_{4,68}$1.96.10) or DIRECT ($_{4,68}$0.32.85). However, there was a small effect on BLOCK for VIEWPORT ($_{4,68}$2.67.03), but corrected post hoc tests did not detect a significant result (all $p \geq .44$). There was no significant effect found in error rate across all BLOCKS. With no strong learning effects present, all blocks were retained in the analysis below.

4.5.2. Error rate

It is possible to make more than one target selection error per trial, but we define *Error Rate* to be the proportion of trials in which one or more errors occurred. For only the trials containing at least one error, the mean number of errors is 1.29 ($\sigma = 0.73$). Overall, direct input is least error prone and using a viewport is most error prone (Fig. 3-right). There is a significant main effect for TECHNIQUE ($_{2,34}$20.32.001) with post hoc tests finding DIRECT has fewest errors (3.3%), followed by RAYCAST (9.3%), then VIEWPORT (12.2%) (all $p > .002$).

Direct input had as few, or fewer, errors than raycasting, while viewport typically had as many, or more, errors than raycasting (Fig. 3-left). A significant interaction between TECHNIQUE and TARGET ($_{3.08,52.46}$21.37.0001) with post hoc tests reveal that for HIGH target types, VIEWPORT (4.5%) has more errors than both RAYCAST (1.4%) and DIRECT (0.7%) (all $p > .035$). For all other target types, DIRECT is significantly less error prone ($p > .01$) with the exception of LOW, likely due to the difficulty of reaching to tap on floor targets. A pronounced difference is for TABLE targets, where DIRECT (2.8%) has an order of magnitude fewer errors than RAYCAST (31.6%) and VIEWPORT (47.2%) (both $p > .001$).

4.5.3. Movement Time

The *Movement Time* is the duration from the moment the start target is selected until the moment the measurement target is selected. We include all trials, regardless of errors. Overall, raycasting is fastest and direct input slowest (Fig. 4-right). There is a significant main effect for TECHNIQUE ($_{2,34}$7.39.002), with post hoc tests finding the difference between each technique significant ($p > .001$): RAYCAST (1.75s) is slightly faster than VIEWPORT (1.89s) and DIRECT (2.10s).

When considering target types, raycasting is fastest for large and high targets, direct input is fastest for targets on the table, while viewport is comparable, or slightly slower, than the fastest technique for all target types, except when targets are on the edge of the table (Fig. 4 left). A significant interaction between TECHNIQUE and TARGET ($_{8,136}$69.59.0001) with post hoc tests finding differences between all techniques and target types (all $p > .03$), except LOW, which had no difference between VIEWPORT and RAYCAST ($p > .41 =$). Highlighting salient results: RAYCAST was fastest for HIGH (1.59s) and LARGE (1.24s) targets, but no significant effect was found between RAYCAST (2.24s) and VIEWPORT (2.11s) for LOW; DIRECT is fastest for both MID (1.38s) and TABLE (2.29s). For targets on the table edge, VIEWPORT is slower than the other techniques with 4.7s on average.

4.5.4. Occluded targets

Our experiment protocol does not strictly control for occluded targets, but the diverse target types we test within a reasonably complex geometric setting of objects and surfaces naturally leads to trials in which there is some visual occlusion of the measurement target. To examine the effect of naturally occurring target occlusion, we create a new independent variable. Whole or partially occluded measurement targets are identified at the moment the start target was selected using the *visibility ratio* metric, calculated from the user's head (see Apparatus section). We use this to create a five-level OCCLUSION factor, with each

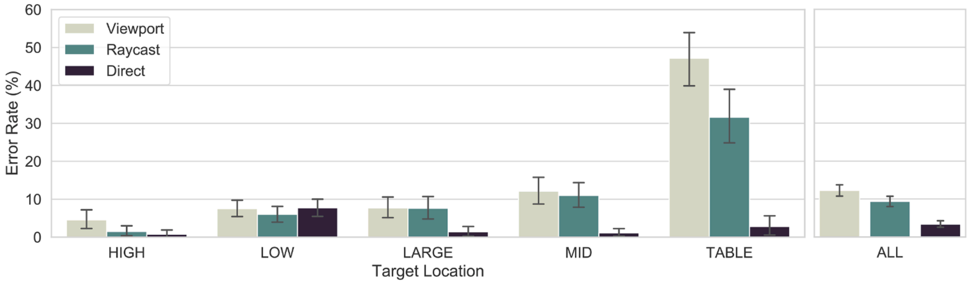

Fig. 3. *Error Rate* for each TECHNIQUE by: TARGET type (left); all target types combined (right). Error bars in all figures are 95% CI.

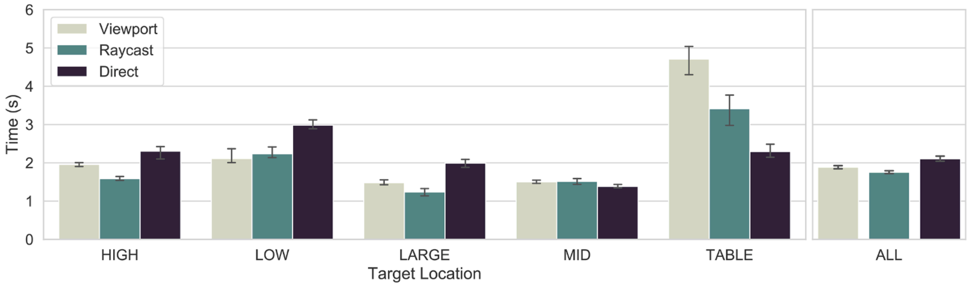

Fig. 4. *Movement Time* for TECHNIQUE and TARGET types (left). MT for TECHNIQUE on all combined types (right).

level representing a 20% bin (see x-axis of Fig. 5).

There is a significant interaction between TECHNIQUE and OCCLUSION ($_{2,5029}$31.78.0001) on *Movement Time* (Fig. 5). Post hoc tests show that target occlusion has no effect on DIRECT input for movement times for the $[0\% - 40\%)$ and $[60\% - 100\%]$ bins ($p \geq 0.32$). In contrast, there is an effect for VIEWPORT and RAYCAST, for which movement time steadily increases over each bin by an average amount of 0.68s and 0.6s respectively ($p > 0.006$).

4.5.5. *Target view angle*

Like occlusion, we create a new independent variable to capture where the surface of a target is pointing relative to the participant's head position. This can be expressed as the *target view angle*, defined as the angle of incidence between a ray beginning from the user's head position and ending at the centre of the target, to the surface normal of the target. Using logged data, this angle is calculated at the moment the start target is selected. We use these angles to create a five-level ANGLE factor, with the first four levels representing 22.5° bins, and the fifth level for all view angles more than 90°, where 0° is when the target is visible and its surface normal is directly facing the participant (see *x*-axis of Fig. 6).

There is a significant interaction between TECHNIQUE and ANGLE ($_{2,5029}$152.65.0001) on *Movement Time* (Fig. 6). We found that the movement time for VIEWPORT and RAYCAST increase over each view angle bin between 0° and 90° with an average increase of 0.21s and 0.35s respectively (all $p > 0.001$). In contrast, the movement time for DIRECT decreases among the same range with an average reduction of 0.26s (all $p > 0.001$). Each technique converges to similar values for the view angle bin $[45^\circ - 67.5^\circ)$, with reported movement times of 1.95s, 2s, and 1.9s for viewport, raycast, and direct respectively.

An angle greater than 90° indicates that the user was behind the target at the start of the trial. Both VIEWPORT (4.91s) and RAYCAST (4.56s) see a large increase in movement time when users are behind a target ($p > 0.001$). In contrast, DIRECT (2.3s) is more robust to large view angles, with no reported difference when compared to view angles between 0° and 45° ($p > 0.64 =$).

4.5.6. *Head movement*

The different target locations and sizes within a reasonably complex geometric setting naturally leads to trials where the participant must move, or chooses to move, as part of their selection action. To examine this, we calculated a dependent variable for *Head Movement*, defined as the summation over the movement path by the participant during a trial.

We found direct input requires four times more head movement overall (Fig. 7 right). The main effect of TECHNIQUE on *Head Movement* is significant ($_{2,34}$391.93.0001), with post hoc tests finding all techniques significant (all $p > 0.003$): DIRECT (138cm); VIEWPORT (34cm), and RAYCAST (28cm).

We also found participants often move to adjust their position to see an initially occluded target, since occluded targets also increase the amount of head movement during a trial (Fig. 7 left). A significant interaction between TECHNIQUE and OCCLUSION ($_{2,34}$42.56.0001) on *Head Movement*, post hoc tests show a significant result among all techniques (all $p > .025$) with the exception of DIRECT where no significance is measured for the $[60\% - 80\%)$ and $80\% +$ groups ($p > 0.35 =$). All techniques demonstrate a positive correlation between distance moved and occlusion, where VIEWPORT increases by 16cm, RAYCAST by 11cm, and DIRECT by 25cm on average.

4.5.7. *Subjective ratings*

After the main experiment was completed, the participant rated each technique from 1 (worst) to 10 (best) for four subjective measures. Data for each was transformed using Aligned Rank Transform (Wobbrock et al., 2011) to correct non-normality, but no main effect for TECHNIQUE was found for any subjective measure. Combined average scores across techniques are 9.1 for *ease-of-learning*, 8.0 for *comfort*, 8.0 for *ease-of-use*, and 7.6 for *overall performance*. We expected direct input to be rated lower due to higher physical effort, but our data does not support this.

4.6. *Discussion*

We found important differences among the three techniques. Direct input may be slower overall for tested conditions, but it also had the

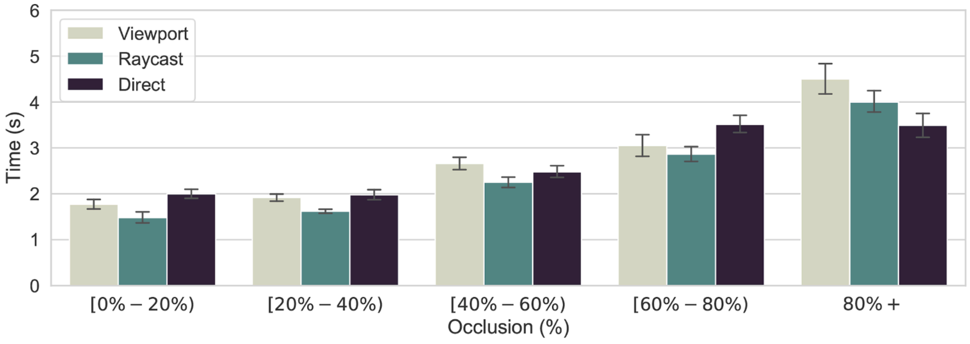

Fig. 5. *Movement Time* by OCCLUSION by TECHNIQUE.

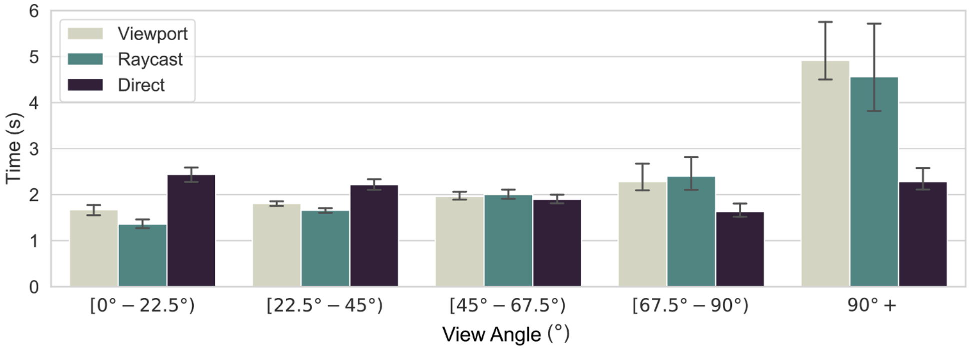

Fig. 6. *Movement Time* by target view ANGLE by TECHNIQUE.

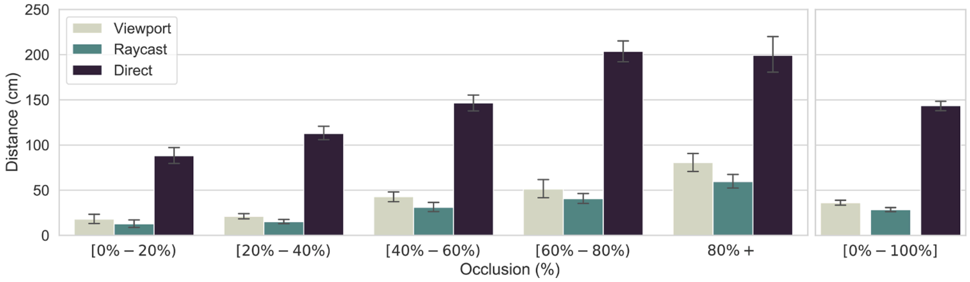

Fig. 7. *Head Movement* by OCCLUSION by TECHNIQUE.

lowest error rate, except for targets near the floor. In some cases, like the targets on the table edge, on the box, or behind the box, direct input was faster and had an order of magnitude lower error rate compared to the other techniques. Perhaps because that particular group of targets where narrower then the others, physical interaction made it easier to control. On the other hand, raycast was fastest overall, and as fast or faster than the other techniques for all target types except in the table group. For the most part, viewport had comparable, or only slightly worse time and error compared to raycast. Notably, viewport was as fast as raycast for targets on or near the floor, possibly due to how the mobile phone's camera naturally points down when holding it.

Overall, our results suggest raycast or viewport are good overall pointing methods in SAR, but direct input should still be considered for small targets that are near arms reach or less. Further, a hybrid technique may also be possible. Analogous to Parker et al.'s TractorBeam, a method that transitions between raycast and direct pen input on a tabletop, a hybrid technique could be designed for mobile phonepointing in SAR using the context of the space and proximity of the user to surfaces. For example, if the phone contacts a surface or object, then a direct input selection is made. Otherwise, raycast or viewport pointing could be used depending on the particular use case of the task. In particular, viewport does not suffer from self-occlusion, so could be used when targets are hidden by the user's shadow and blocking a projected image from being seen (Hartmann and Vogel, 2018).

The effects of target occlusion and view angle on movement time, and differences in head movement distance, especially to compensate for occlusion, suggests these are important factors affecting pointing time in SAR. In the next experiment, we strictly control these factors to better understand their effect.

5. Experiment 2: simulated SAR pointing task

The goal of this second experiment is to validate results of Experiment 1 in a more controlled SAR pointing task. To achieve high control over target placement, occlusion, and view angle, we simulate specific conditions of a SAR pointing task by rendering targets and occluding geometry in an AR HMD. The pointing context under investigation still remains surface mappedSAR since the targets are 2D, just as they would be if mapped onto real 3D surfaces. We test a reduced range of target distances compared to Experiment 1, this decreased the number of factors making the study practical to run, but it does mean our results are more representative of a best case task in terms of reach.

Using an AR HMD is much more practical and flexible than actuating the physical environment itself (Cheng et al., 2018), or creating a physical layout of real objects and targets with constraints for the participant's initial position. Simulating SAR in AR enables target consistency across a diverse set of participants: we can place targets and objects around the user so that the distances, height, occlusion, and size are exactly the same for each participant regardless of their height or where they stand. There are limitations to this approach, the field-of-view of the HMD is smaller than the human eye's, wearing an HMD can be uncomfortable and requires a tether attached to a computer, and there is no natural tactile feedback in the direct condition. However, we took steps to minimize these aspects by using a wide 90° field-of-view AR HMD, the HMD is light since it is tethered eliminating the need for heavy batteries, we were careful routing the HMD tether to avoid obstruction, and we used phone vibration to simulate physical surface contact with the direct technique. We discuss these limitations again in Section 7.5.

5.1. Participants

We recruited 12 participants, ages 19–28, 10 male, 2 female, 10 were right-handed. Overall, they reported using a mobile device an average of 3.5 h a day. Participants received $15 for their time. This experiment was conducted 2 months after Experiment 1, and no participants participated in both experiments.

5.2. Apparatus

The Unity3D software running on the server was modified to render targets and geometry to a Meta2 AR HMD (2550 × 1440 px, 90° FOV), which is tracked with the Vicon which ensures that targets and visuals are precisely placed and remain stable relative to HMD movement.

Meta2 depth compression (a known issue with the headset at that time) was corrected to simulate a real world view by applying a logarithmic function to the target and occluding geometry positions. The room was empty, neutral, and clear of unnecessary clutter. All SAR environment surfaces and targets are rendered in the HMD and illuminated to ensure easy identification in the environment. Using rendered virtual targets means there is no physical feedback in the direct technique. We vibrate the phone when it contacts a virtual surface to compensate. These considerations combine to make perception of the task in AR reasonably similar to SAR.

The same Pixel phone was used, and in all conditions, the real phone screen was used for input and output (there is no virtual overlay). For example, in the viewport technique, the actual phone screen renders a view of the same controlled 3D geometry (obstructions and targets) used to render the AR HMD. The rendering simulates what would be seen from the phone's real back camera.

5.3. *Task*

The task was to select two targets in sequence as quickly and accurately as possible. The first target was a circular *start target* (r = 18cm) located at a fixed position directly in front of the user, 150 cm above the floor, oriented towards them. The second target (the *measurement target*) was a red circle (r = 13cm). To increase task variability, these targets were placed at different positions relative to the start target. They were distributed on the surface of a hemisphere into 9 radial positions (30° intervals) from a point of origin (the user's head position) at a "near" and "far" distance (67 and 124 cm) relative to the origin like two concentric spheres (Fig. 8a). The near distance was chosen to be within arms reach and the far distance requires some body movement to reach. Before the start of the trial, both the start target and measurement target where displayed to the user before occlusion to ensure that the trial accurately reflected a selection task and not one where the participant needed to first search for the target. Using a start and measurement target is reminiscent to early works in Fitts' Law analysis from which our inspiration was derived (Meyer et al., 1988).

These varied target positions generalize our results when considering the primary factors of occlusion and target view angle. The targets are rendered in midair to simplify the scene and avoid unnecessary rendering, but they are still 2D as though they were mapped into a 3D surface. What is important is their position relative to participant.

5.4. *Design and protocol*

The design is fully within-subjects. The primary independent variables are TECHNIQUE (3 levels: VIEWPORT, RAYCAST, DIRECT), target OCCLUSION (2 levels: 100% occluded, 0% occluded), and target view ANGLE (2 levels: 0°, 90°). A target view angle of 0° means the normal of the target points towards the participant and the full target is easily viewed if not occluded. A view angle of 90° means the target normal is orthogonal to the participant's view, and the target appears as a thin slice until the participant adjust their head position. To control target occlusion, a large grey wall was rendered between the participant and the target to create the desired occlusion level (Fig. 8b). Target view angle was controlled by rendering the target normal at the desired angle relative to the participant. The ordering of TECHNIQUE was counter balanced using a Latin square. For each TECHNIQUE, the participant completed 3 BLOCKS of trials presented in random order.

The instructions, technique practice, and post experiment questionnaire were the same as Experiment 1. Participants were free to move around the space, but were required to return to a starting position at the beginning of each trial. The entire session lasted approximately 60 minutes. In summary: 3 TECHNIQUES × 3 BLOCKS × 2 OCCLUSION levels × 2 ANGLE levels × 17 target positions (8 near and 9 far), resulting in 612 data points per participant.

5.5. *Results*

The same analysis methods from Experiment 1 are used. Similar to the first experiment, 133 (1.8%) outliers were removed.

5.5.1. *Learning effect*

There is a significant BLOCK × TECHNIQUE interaction on *Movement Time* ($_{1.35,25.71}$30.91.0001), but not on *Error Rate*. Post hoc tests found block 1 significantly slower than blocks 2 and 3 (both p > .0001), suggesting a learning effect in block 1. In all subsequent analysis, we use only blocks 2 and 3 for the best estimation of practised performance.

5.5.2. *Error Rate*

There is a significant effect of TECHNIQUE on *Error Rate* ($_{2,22}$7.50.01). Overall, raycast is least error prone (4%), direct input is the most error prone (11%), and viewport falls in between (9%).

There is a significant effect of TECHNIQUE × OCCLUSION on *Error Rate* ($_{2,22}$10.60.001). A post hoc analysis shows that DIRECT is least

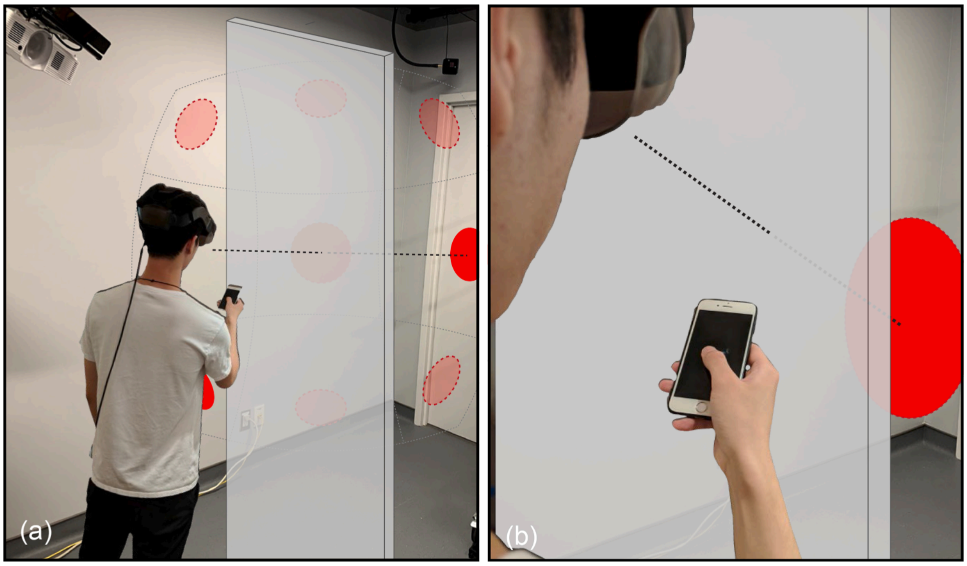

Fig. 8. Illustration of the Experiment 2 simulated SAR pointing task using an AR HMD and real phone: (a) near target positions spatially distributed around the user; (b) participant point of view showing partially occluded target. Note the real phone screen was used for input and output (there is no virtual overlay).

error prone when targets are non-occluded (1.8%) and most error prone when occluded (20.3%). In contrast, the error rate for both VIEWPORT and RAYCAST remained the same across occlusion levels with no significant effect ($p \geq 0.48$).

There is a significant effect of TECHNIQUE × ANGLE on *Error Rate* ($_{2,22}$5.670.01). Post hoc tests show an effect of ANGLE on VIEWPORT ($p > 0.001$) where the error rate is 6% without rotation and 12% when rotated. In contrast, there is no reported effect of ANGLE on RAYCAST or DIRECT.

5.5.3. Movement time

Overall, direct input is fastest and raycast is slowest. There is a significant main effect of TECHNIQUE on *Movement Time* ($_{2,11}$11.70.001), with post hoc tests finding a significant effect among all techniques ($p > .034$): RAYCAST (2.03s) is slightly slower than VIEWPORT (1.92s) and DIRECT (1.69s).

When considering occlusion and angle factors, viewport is fastest for far targets with the best view angle, direct input is fastest for near targets that have poor viewing angle, while raycast is comparable (or slightly slower) than viewport for far targets with the best view angle (Fig. 9). A significant interaction between TECHNIQUE, OCCLUSION and ANGLE ($_{2,22}$21.61.001) and post hoc tests found varying differences between techniques, occlusion, and angle target. Highlighting the most salient results: DIRECT was fastest for all near and non-rotated targets at 1.14s ($p > .001$), but both RAYCAST (1.57s) and VIEWPORT (1.56s) are essentially tied for far, non-rotated, and non-occluded targets.

5.5.4. Subjective ratings

After the main experiment was completed, the participant rated each technique from 1 (worst) to 10 (best) using four subjective measures: ease-of-learning, comfort, ease-of-use, and overall performance. There was a significant main effect of TECHNIQUE on *ease-of-learning* ($_{2,22}$9.450.01), with post hoc tests finding that direct input was perceived easier (9.0) compared with viewport (6.4) and raycast (7.75). No other subjective measures had significant effects.

5.6. Discussion

We found similarities and differences with Experiment 1. Although direct input only had simulated haptic feedback when contacting virtual targets, it still outperformed both raycast and viewport for near and rotated targets. We observed the relative robustness of direct input to rotated targets, with movement time across rotation remaining similar. However, the performance increase for direct could be partly the result of how we structured our experiment. Since our setup creates virtual walls and surfaces, the participant did not have to slow down when hitting the target like they would with a real surface, allowing them to keep their velocity and partially “punch through” the virtual wall to hit targets. People may be unlikely to strike a real surface with a phone using the same speeds and forces. With viewport and raycast, target view angle has a pronounced effect: viewport performed best for far non-rotated targets, while raycast was in-between. This contrasts with Experiment 1, where raycast was fastest with more varied target situations.

Both experiments reveal useful insights into the three pointing techniques under investigation. Experiment 1 provides a more authentic setting, which is complimented by the carefully controlled Experiment 2. Together they provide a more holistic view into how each technique performs under different SAR environment settings. To further synthesize and generalize these empirical results, the next section describes a unified analytic model of SAR pointing.

6. Modelling a SAR pointing task

To further validate, understand, and generalize the results of both our experiments, we develop a model to predict the time of a general SAR pointing task when using a mobile phone in a potentially occluded environment. A model is useful to validate our empirical observations that target occlusion, target viewing angle, and user movement are primary contributing factors in pointing motions. A model helps us understand characteristics about the pointing motion itself, such as whether Euclidean or angular distance is more dominant, and to reveal relative trade-offs in pointing techniques. Finally, a model can be used by designers of future SAR environments to predict approximate task times and make informed decisions about where to place content.

We base our model on the data collected from the previous two experiments and on previous investigations into the empirical applications of Fitts’ law (MacKenzie, 1992). It incorporates key aspects of pointing in a SAR environment: how a user has to move to select the target, how occluded the target is relative to the user, and the visual angle of the target. Using data collected from Experiments 1 and 2, we develop a model by first analyzing how each data attribute (e.g., distance, occlusion, angular width, etc) contribute to the observed movement times across trials. We then look at previous Fitts’ law pointing models found

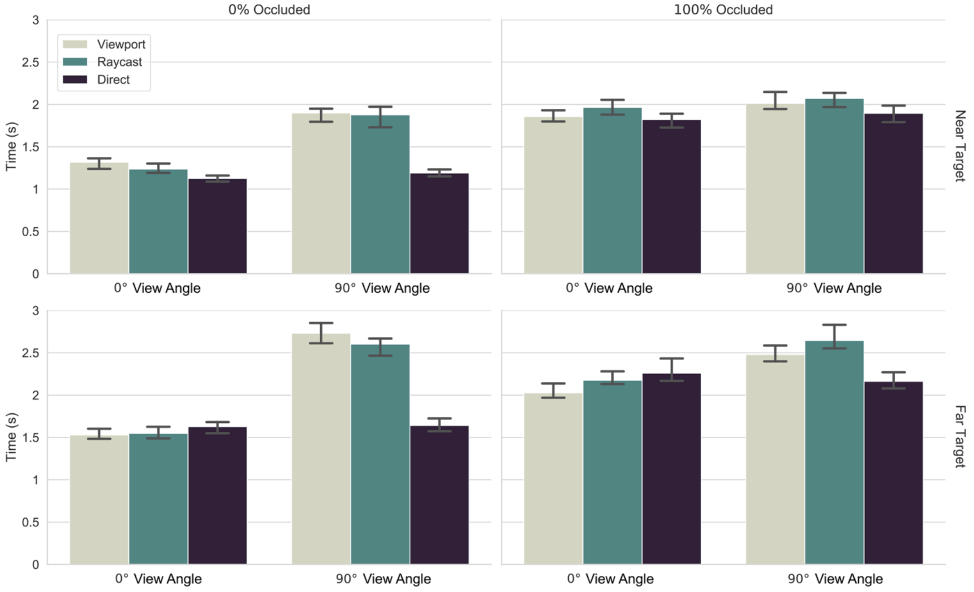

Fig. 9. *Movement Time* by TECHNIQUE by target view ANGLE for combinations of target OCCLUSION and DISTANCE.

in the literature, and using separate datasets from Experiments 1 and 2, we test these previous models in a comparison with three variations of our general model formulation using different norms. The results show our proposed model is a much better predictor than previous Fitts' law formulations across all three phone pointing techniques.

6.1. Previous pointing models

The objective of all models is to predict a movement time *MT* based on key parameters of a pointing task. MacKenzie and Buxton (1992) were among the first to extend Fitts' law into 2D using a "min" model (Eq. (1)), then Accot and Zhai (2003) improved on this with their weighted ℓ_2-Norm model (Eq. (2)):

$$MT_{min} = a + b\log_2\left(\frac{A}{\min(W, \alpha_1 H)} + 1\right) \tag{1}$$

$$MT_{Wt\ell_2} = a + b\log_2\left(\sqrt{\left(\frac{A}{W}\right)^2 + \alpha_1\left(\frac{A}{H}\right)^2} + 1\right) \tag{2}$$

The pointing task parameters used in these two models are *A* for target "amplitude" (the distance to the target), and *W* and *H* for the width and height of the target. These are regression models, so other parameters in the model formulation are related to the regression fitting (*a* for the intercept and *b* for the slope) and the relative weighting of the terms (α_1).

Subsequent extensions of Fitts' law into 3D were focused on pointing at fully 3D targets, such as a rectangular cuboid (six quadrilateral faces forming a convex polyhedron) rendered on a 3D volumetric display (Grossman and Balakrishnan, 2004). Their proposed trivariate model considered all three spatial dimensions of the target (*W*, *H*, and *D*) where the resulting model formulation used regression fittings across 5 parameters, which is similar to our proposed formulation. Cha and Myung (2013) later expanded on these ideas using 2D squares suspended at arbitrary locations in 3D spherical coordinate space.

Pointing in surface mappedSAR arguably has more in common with 2D selection, like the "min" and ℓ_2-Norm models above. Although there is a 3D environment, all targets are, in fact, flat 2D objects on 2D surfaces, and any feedback (like a cursor) is restricted to the same 2D surfaces. Relevant to this, both Jota et al. (2010) and Kopper et al. (2010) proposed a modified Fitts' model for raycast pointing at 2D targets rendered on a large display (Eq. (3)). It takes into account the angular target width (W_ω) and angular distance to the target (D_δ) for improved predictability. A nice property of angular distances and widths is that they are invariant to scale.

$$MT_{\text{Angular}} = a + b\log_2\left(\frac{D_\delta}{W_\omega} + 1\right) \tag{3}$$

A traditional single-term Fitts' model, like Eqs. (1)–(3), have a nice property that there is a clear "index of difficulty" for the pointing task in the form of log_2 term. Such single term models have been used to model a small table-top SAR environment where there is no occlusion, no user movement, no extreme target angles, and the entire surface topology can be projected onto a 2D surface without any overlap (Gervais et al., 2015). However a simple single term model will not capture the complexity of pointing with a phone in a surface mappedSAR environment with varied surface geometry.

Rohs and Oulasvirta (2008) developed a two-term Fitts' pointing model that separates two stages of viewport AR pointing for near-planar targets (2008) and at-distant targets (Rohs et al., 2011): first move the phone to frame the target, then use the screen for detailed adjustment. Their resulting model is a linear combination of the two corresponding movement times. Their proposed two-term Fitts' Law approach inspired our general Fitts model formulation for viewport selection. However, they model a slightly different viewport selection technique, where the viewport acts more like a telescopic sight with a cross-hair centre. Further, their target context does not capture a complex geometric environment where target occlusion and varying target view angles are present. As such, a direct adaptation of their model is not possible.

6.2. Surface mapped SAR pointing model

In surface mappedSAR, our experiment results suggest there are multiple aspects that affect movement time, such as occlusion, target angle, and user movement. Similar to Rohs and Oulasvirta, these can be combined into multiple movement stages to create a multi-term model. This increased model complexity is an artifact of mapping 2D targets to more complex 3D geometry. Our model was designed in two stages. First we identified the key parameters in a SAR mobile phonepointing task, then we use the most important parameters to design a two-term model.

6.2.1. Model parameter selection

We used a common machine learning methodology practised in empirical software engineering to determine the most salient features for a given measure (Gousios et al., 2014; Ray et al., 2014). In our case, we determine salient features in the prediction of movement time (MT). We analyzed the data from both experiment datasets: a total of 42 candidate features were identified and computed from every experiment trial. These included various forms of target distances, orientations, widths, and heights; various expressions of target occlusion; various forms of measuring user movement during the trial; as well as initial and final positions of the user relative to the target. Where possible, we included parameters in both angular and Euclidean form.

The method worked as follows. Experiment trials were split into 90% training and 10% testing sets using random sampling. A random forest (RF) regressor (Geurts et al., 2006) was trained to estimate each trial time using all 42 features. This resulted in an absolute mean error (MAE) of 371ms with an accuracy of 80.3%. From the RF model, we examined feature importance in terms of weights to determine which features were most salient when predicting MT (Table 1). We found that absolute user movement, target occlusion (as a percentage of target visibility, and as a binary variable), target orientation relative to the user, and target angular width, height, and distance were assigned high weights by the RF. This general pattern supports our observations from Experiments 1 and 2.

6.2.2. Model formulation

Our model uses the salient features from the RF regression analysis as parameters in two terms with formulations inspired by Accot and Zhai (2003)'s weighted ℓ_2-Norm model and the angular models proposed by Jota et al. (2010) and Kopper et al. (2010) Similar to how the Rohs and Oulasvirta (2008) model is a linear combination of two terms representing the time during each stage of a pointing action, we use a linear combination of two terms to capture the relative contributions of movement time caused by the spatial configuration of user and target in the SAR pointing task:

Table 1
Features selected by Random Forest regression for MT prediction. Only the top 9 features with weights 0.1 or more are shown, there are 33 other features with lower weights.

Feature	Weight
Integrated Movement (sum over movement path)	0.16
Target Occlusion (as % of target visibility)	0.07
Target Orientation	0.06
Target Angular Width/Height	0.04
Target Occlusion (as binary classification)	0.04
Distance between Start and End Positions (Euclidean)	0.03
Target Angular Distance	0.02
User Angular Movement	0.02
Target Distance from Start (Euclidean)	0.01

$$MT_{\mathrm{SAR}} = MT_1 + MT_2 \tag{4}$$

where MT_1 captures the movement of the user during the pointing task, and MT_2 captures the final spatial relationship of the user and the target. Eq. (5) is the complete expression of the generalized ℓ_p norm for the two-term model:

$$\begin{aligned} MT_{\mathrm{SAR}\ell_p} = a + b\log_2\left(\left(\left|\frac{M_\delta}{W_\omega}\right|^p + \alpha_1\left|\frac{M_\delta}{H_\omega}\right|^p\right)^{\frac{1}{p}} + 1\right) \\ + c\log_2\left(\left(\left|\frac{D_\delta}{W'_\omega}\right|^p + \alpha_2\left|\frac{D_\delta}{H'_\omega}\right|^p\right)^{\frac{1}{p}} + 1\right) \end{aligned} \tag{5}$$

In each term, the model expresses relationships between angular distance and angular width, or angular distance and angular height, as ratios. This is similar to traditional 2D models using Euclidean target amplitude over width (A/W). The parameters contained in each term of the model, and associated features used in their calculation, are illustrated in Fig. 10 and explained as follows:

- MT_1 captures the movement of the user based on their angular movement M_δ and their angular width W_ω and height H_ω from position P to position P' relative to the measurement target.
- MT_2 encapsulates the configuration of the user and targets after movement. This includes the ending angular target distance D_δ, width W'_ω, and height H'_ω, all relative to user's ending position P'.

Similar to the analysis conducted by Accot and Zhai (2003), we evaluate our model by comparing the observed and predicted movement times across three norm variations stemming from Eq. (5): the ℓ_1-norm (when $p = 1$), the ℓ_2-norm (when $p = 2$), and the ℓ_∞-norm (when $p = \infty$).

We now provide more justification and explanation for the terms and parameters.

6.2.3. *Representing user movement*

Given that a user moves dynamically in the environment, a unique challenge is how to properly model target distance. Traditional models, like Accot and Zhai (2003)'s, use the distance from previous target to the current target as movement amplitude A. The model from Jota et al. (2010) and Kopper et al. (2010) uses the same target to target movement, but using angular distance D_δ. This may make sense when the target is visible or no major user movement is required, but this fails for tasks in which the user can move freely in the space or the target is occluded by an obstacle.

In our key parameter analysis, we found the user's *integrated movement* (sum of all movements over the entire movement path) to be the top feature regardless of technique, but this is an empirical measure calculated during the experiment. To find a representation of user movement that could be calculated for an arbitrary pointing task a priori, we analyzed relationships between the angular widths and distances between the user's start and end positions with respect to their calculated integrated movement. We found the product between the angular movement M_δ and the ratio W'_ω/W_ω to have the highest correlation for target width (and similarly between M_δ and H'_ω/H_ω for target height). This was even higher than other measures, such as the straight line Euclidean distance (M) between the start and ending positions of the user.

Following Jota et al. (2010) and Kopper et al. (2010), we use this new approximation for movement in the numerator, and W'_ω in the denominator for width (and similarly H'_ω for height). This means the movement M_δ is scaled by the ratio W'_ω/W_ω (and H'_ω/H_ω), to account for movement towards or away from the target, and then subsequently divided by the resulting target angular width W'_ω (and height H'_ω) as done by Jota et al. (2010) and Kopper et al. (2010). This simplifies to simply M_δ/W_ω (or M_δ/H_ω) as shown in the MT_1 term of Eq. (5) as follows:

$$\frac{M_\delta\frac{W'_\omega}{W_\omega}}{W'_\omega} = \frac{M_\delta}{W_\omega} \quad \text{and} \quad \frac{M_\delta\frac{H'_\omega}{H_\omega}}{H'_\omega} = \frac{M_\delta}{H_\omega} \tag{6}$$

In practical terms, this means that as the angular width decreases, this relationship places more importance on the required user movement, increasing the contribution of this ratio measure. Estimating the

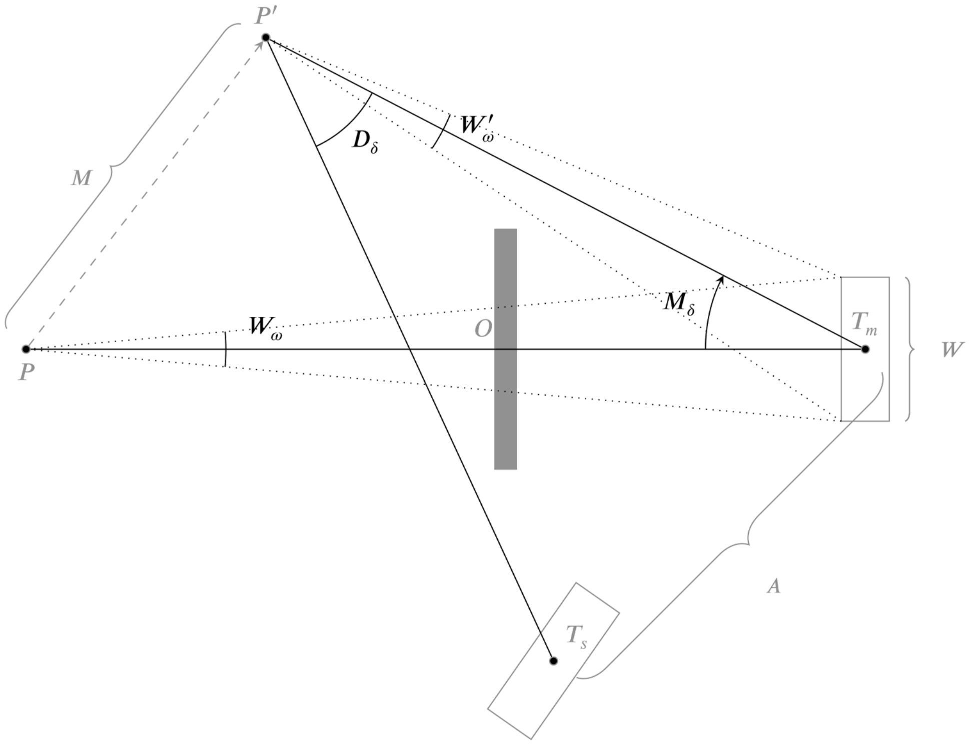

Fig. 10. Illustration of a generic surface mappedSAR pointing task in a simplified 2D top-down view. There is a start target T_s, a measurement target T_m, an obstacle O causing initial measurement target occlusion, an initial user position P at the start of the task, and an ending user position P' at the end of the task. Target distance in our model is represented as the ending angular distance D_δ from start target to measurement target relative to the user's ending position. Other models use absolute Euclidean distance between the two targets, shown as amplitude A. Our model uses the user's angular movement M_δ required to avoid occlusion, which is a scale invariant form of the Euclidean absolute user movement M. Our model uses the initial angular target width W_ω and the ending angular target width W'_ω relative to the user at those respective times. Other models use the absolute width W. Angular target height H_ω and absolute target height H are not shown, but calculated similarly.

M_δ parameter when predicting the pointing task time can be accomplished by first analyzing the 3D geometric relationships between the target, any occluding objects, and the user's initial position: M_δ is then the angular movement required to make the target completely unoccluded.

6.2.4. *Representing target view angle*

The geometry of a SAR environment can be complex, so the geometry and surface mappedtargets viewed from the user's point-of-view also inherit this complexity. One aspect of this is the view angle of the target with respect to the user's gaze (also called the "target incident angle"), which can vary dramatically. To incorporate this, and to combine the related parameters of target width and view angle, we calculate the target angular width along the x-axis and y-axis as viewed by the user at the starting position P and again at the ending position $P^{'}$ during the pointing task. This is similar to the angular distance and width used in the Fitts' law models proposed by Jota et al. (2010) and Kopper et al. (2010). We denote these as W_ω and H_ω for the initial starting width and height at position P and $W_\omega^{'}$ and $H_\omega^{'}$ for the ending width and height at position $P^{'}$.

6.2.5. *Representing target occlusion*

In our study, and from the key parameter analysis (Table 1), occlusion has a significant impact on movement time across all techniques. To account for this, we formulate occlusion using the spatial and geometric configuration of the user, the targets, and the obstacles placed within the SAR environment. These components describe the actions required by the user to go from a state of target occlusion to increased target visibility. In our model's formulation, this is represented by term MT_1, where the ratio between the angular movement M_δ and angular width W_ω captures the movement required when a target is occluded by an obstacle O when the user is initially at position P.

6.3. *Model evaluation*

We compare the three norm variations of our SAR pointing model, which we refer to as *SAR* ℓ_1, *SAR* ℓ_2, and *SAR* ℓ_∞, with applicable models from previous work, MacKenzie and Buxton (1992)'s *Min* model, Accot and Zhai (2003)'s *Weighted* ℓ^2 model, and the angular model proposed by Jota et al. (2010) and Kopper et al. (2010).

6.3.1. *Method*

All models were evaluated for each phone pointing technique (raycast, viewport, and direct) using the combined experiment datasets: Experiment 1 data captured in a realistic SAR environment, but with less strictly controlled spatial pointing conditions; and Experiment 2 data with highly controlled pointing conditions using a simulated SAR pointing task facilitated by AR HMD.

The evaluation consisted of non-linear least squares optimization with respect to the regression parameters (a, b, and c) and the weight parameters (α_1 and α_2) as they are used in each of the evaluated models. For example, Mackenzie and Buxton's *Min* model and Accot and Zhai's *Weighted* ℓ^2 model use only parameters a, b, and α_1, while our *SAR* models use all parameters: a for the overall intercept, b and c for the slopes of each term, and α_1 and α_2 for the weights of each term. This results in 5 parameters total, which is the same number of parameters used in Grossman and Balakrishnan (2004) trivariate pointing model. Movement time is discretised into increments of 250ms for all models using the mean in that interval across all participant data. This is the approach Gervais et al. (2015) used for their analysis of table-top SAR pointing.

6.3.2. *Results*

The fitted model parameters are listed in Table 2 for Experiment 1 and Table 3 for Experiment 2. Adjusted R^2 is used as the primary basis for comparison.

Across both experiment datasets, the three different formulations of the *SAR* model have higher Adjusted R^2 fitness metrics compared to the previous models. For Experiment 1 data, the SAR models do not capture direct pointing as well as the other two techniques (Adjusted R^2 of 0.22 to 0.73 for direct, compared to Adjusted R^2 of 0.85 to 0.96 for raycast and viewport). For the more controlled, and more restricted SAR pointing tasks in Experiment 2, all techniques have high fitness (Adjusted R^2 are all equal to or greater than 0.88).

The accuracy of the three *SAR* models are further visualized in Figs. 11 and 12. The diagonal grey line represents the line of equality between measured and predicted MT, a perfect model would have all points lie on this line. The two figures are similar in terms of the distribution of points. There is an increase in confidence interval size as MT increases, likely related to how the models handle uncertainty in the more diverse target selection tasks requiring more MT.

For Experiment 1 data, the alpha weights heavily favour the angular heights compared to the angular widths: α_2 in particular has very high

Table 2
Model fittings using Experiment 1 data.

Model	Technique	a (ms)	b (ms/bits)	c (ms/bits)	α_1	α_2	R^2	Adj R^2
Min	Raycast	1.95	4.21		$< .01$		0.02	-0.01
Min	Viewport	1.90	3.92		$< .01$		0.00	-0.02
Min	Direct	1.87	3.69		$< .01$		0.09	0.07
Weighted ℓ_2	Raycast	552.77	431.54		0.39		0.62	0.61
Weighted ℓ_2	Viewport	455.79	489.39		1.40		0.67	0.66
Weighted ℓ_2	Direct	924.88	317.96		20.98		0.09	0.08
Angular	Raycast	1643.89	223.60				0.47	0.44
Angular	Viewport	1663.75	298.20				0.24	0.20
Angular	Direct	2190.26	40.11				0.05	0.01
SAR ℓ_1	Raycast	978.45	320.25	176.18	27.81	$< .01$	0.97	0.96
SAR ℓ_1	Viewport	3.64	563.47	84.69	2.36	137281	0.86	0.85
SAR ℓ_1	Direct	1725.92	331.43	-243.44	52.38	22.13	0.37	0.32
SAR ℓ_2	Raycast	1121.89	349.18	60.62	224.96	22.72	0.93	0.93
SAR ℓ_2	Viewport	1302.06	448.40	72.95	48.21	$< .01$	0.94	0.94
SAR ℓ_2	Direct	1730.64	346.64	-258.39	857.28	110.78	0.28	0.22
SAR ℓ_∞	Raycast	501.32	367.75	35.75	12.65	6380206	0.92	0.92
SAR ℓ_∞	Viewport	815.80	472.78	27.30	6.11	3892559	0.91	0.90
SAR ℓ_∞	Direct	1970.77	320.04	-227.61	98.18	124.03	0.74	0.73

Table 3
Model fittings using Experiment 2 data.

Model	Technique	*a* (ms)	*b* (ms/bits)	*c* (ms/bits)	α_1	α_2	R^2	Adj R^2
Min	Raycast	1.47	2.49		< .01		0.25	0.23
Min	Viewport	1.48	2.51		< .01		0.17	0.14
Min	Direct	1.45	2.41		< .01		0.29	0.27
Weighted ℓ_2	Raycast	1099.70	293.91		1.09		0.10	0.07
Weighted ℓ_2	Viewport	912.14	245.46		8.26		0.08	0.05
Weighted ℓ_2	Direct	704.41	289.74		2.65		0.53	0.51
Angular	Raycast	1517.62	245.53				0.00	-0.05
Angular	Viewport	1318.77	282.83				0.35	0.32
Angular	Direct	880.74	374.23				0.26	0.22
SAR ℓ_1	Raycast	56.83	226.09	211.04	79.72	< .01	0.89	0.88
SAR ℓ_1	Viewport	81.04	186.05	139.39	240.18	< .01	0.91	0.90
SAR ℓ_1	Direct	11.06	305.81	248.27	7.68	< .01	0.93	0.93
SAR ℓ_2	Raycast	420.62	253.78	234.25	181.23	< .01	0.91	0.90
SAR ℓ_2	Viewport	352.27	280.40	272.00	46.86	< .01	0.95	0.94
SAR ℓ_2	Direct	-676.70	382.63	258.00	< .01	1421.42	0.94	0.94
SAR ℓ_∞	Raycast	413.40	205.28	189.34	30.63	1.59	0.94	0.93
SAR ℓ_∞	Viewport	597.31	322.77	269.54	2.64	0.13	0.98	0.98
SAR ℓ_∞	Direct	359.21	357.23	333.17	1.47	0.33	0.94	0.94

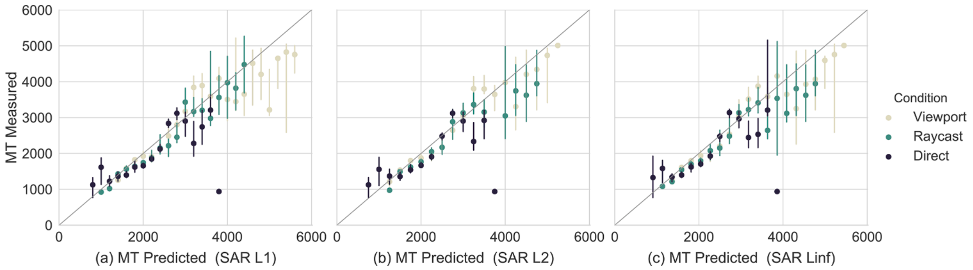

Fig. 11. Measured vs. predicted MT for models for Experiment 1 data: (a) *SAR* ℓ_1; (b) *SAR* ℓ_2 (c) *SAR* ℓ_∞.

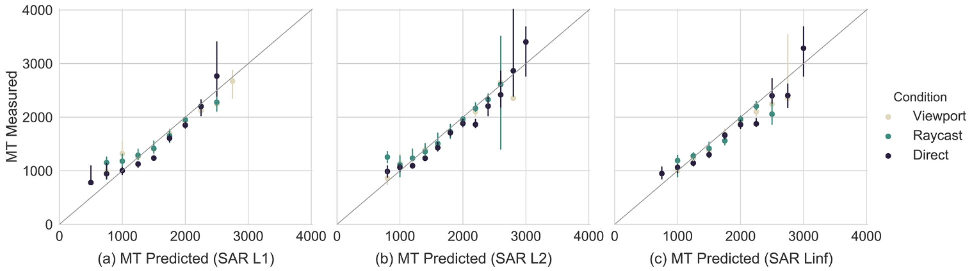

Fig. 12. Measured vs. predicted MT for models for Experiment 2 data: (a) *SAR* ℓ_1; (b) *SAR* ℓ_2 (c) *SAR* ℓ_∞.

relative values for viewport in the *SAR* ℓ_1 model, and for both viewport and raycast in the *SAR* ℓ_∞ model. For Experiment 2 data, the pattern of alpha weights more heavily favour the angular widths in the *SAR* ℓ_1 and *SAR* ℓ_2 models, with the exception of direct input in the *SAR* ℓ_2 model.

The other parameters, namely *b* and *c*, describe the slopes of each term in the model that represent ms/bit (see Tables 2 and 3). We observe that the *b* parameter is larger then *c* across all techniques, which indicates that the first term in our model (MT_1) that encapsulates user movement is more dominate than the second term (MT_2) that encapsulates the pointing task after movement. This correlates with our observations and the Random Forest feature analysis, further providing evidence for the role movement has when selecting targets in SAR.

Looking specifically at direct, in Table 2 for Experiment 1, the negative *c* parameters are uninterruptible and somewhat problematic in the traditional sense of a Fitts' model. This is likely due to how direct input is constrained by the user's proximity to the targets which is not present for distance pointing techniques like raycast and viewport. It is likely the case that since direct input is dependent on environment scale, another model specific for direct pointing would be required to overcome and account for its unique limitations and idiosyncrasies.

7. Discussion

The results of the two experiments, combined with the model analysis, lead to overall findings and design implications.

7.1. Direct input performs well when a target is nearby

The good performance of the direct technique in several target conditions indicates this type of absolute direct input is well suited to SAR when targets are within close proximity to a user. This can be seen in Experiment 1 for the MID and TABLE target types. For targets in

Experiment 2, direct outperforms the other two techniques in most cases, which is different than the pattern in Experiment 1 results. This may be explained by the lack of physical surfaces the user would typically need to navigate, letting them maintain velocity and moving through the virtual barriers without the cost of damaging the mobile phone.

There are other apparent disadvantages to the direct technique that are not present in the distant pointing methods to consider as well, like how much movement is required when targets are far away. This raises questions regarding the suitability of direct selection in large environments, in which the selection cost increases the further the target is away from the user's initial position.

Some other issues are present for direct, which can be seen in the model's regression analysis. In the model parameters for Experiment 1 (Table 2), the direct technique's R^2 values are low relative to the raycast and viewport, and the c coefficients are negative indicating a negative gradient for term 2 in the model. This is undesirable, and indicates that direct selection may require a more tailored modelling equation that better accounts for expected user movement given target proximity, and perhaps variation in the user's arm length. Further exploration of this would be an interesting direction for future work.

7.2. *Viewport affordances*

In the discussion for Experiment 1, we were cautious to recommend viewport overall. Except for some subjective preference of certain target types, there was no clear reason to choose it over raycast or direct input for a given target context. However, Experiment 2 shows the robustness of viewport for different target types in this more homogeneous target setting. During both experiments, we observed that some participants appeared to be reluctant to adjust their physical proximity when using viewport, and would rather attempt selection even if the target was not optimally viewed by the phone camera. The results was an action of repeated (and rapid) touch selection attempts creating the high error rates for the thin table edge targets (i.e. TABLE target type) in Experiment 1. One unique aspect of viewport is in its ability to overlay additional personal or contextual digital information on top of the SAR environment. Though we do not explore this explicitly, it is interesting to note the possible affordances a public and glasses-free SAR environment could have when combined with different *personal viewports*, all occupying the same SAR space. Interesting use cases include multi-person gaming, remote and co-located collaboration, and content sharing. We leave this as another possible direction for future work.

Overall, each technique has advantages and disadvantage when used in a more geometrically complex and large SAR environment. Depending on the context of the task and properties of the target relative to the user, various combinations of raycast, viewport, and direct techniques can be used to accommodate specialized content selection scenarios.

7.3. *Modelling spatial interaction*

Modelling 3D spatial interaction is challenging, and our proposed model is the result of multiple design iterations, experimentation, and hypothesis testing. An early exploration of our model looked at the selectable area of the target after occlusion. However, we found that modelling the selecatable area was less accurate and did not fit within our observations of user behaviour. We found that users would opt to move their head to reveal the object completely than try to select only the exposed target area. This emphasis on head movement can also be seen in our analysis of salient predictors in Section 6.2.1. Our resulting model utilizes these observations to construct a simplified geometric construct of the pointing task, taking into account both head movement and target selection. Another possible approach to modelling movement time is to use a sub-task model like the one proposed by Meyer et al. (1988). Their two-part ballistic and correction model takes into account the velocity and neuromotor noise introduced through the pointing task, which could be an interesting area to explore for future work.

We evaluated our model (Eq. (5)) using three different norms across two different datasets, and based on our results, the R^2 fitness for raycast and viewport are high across all experiments. If we disregard direct input, which we note above may be hard to model due to increased user movement, we can see that the distant pointing techniques perform well when using the ℓ_2-norm ($p = 2$). Though the results from Experiment 2 do suggest that the ℓ_∞-norm ($p = \infty$) outperforms the ℓ_2-norm by a small margin for these techniques, we are hesitant to recommend it given the undesirable large coefficients for α_2 in the Experiment 1 data. Thus, similar to the observations from Accot and Zhai (2003), we recommend using the ℓ_2-norm as the most generalizable version of our SAR pointing model. This version achieved a balanced performance across both datasets.

Previous work explored spatial pointing in full-coverage displays for mouse and raycast in a simple cubemap-like display around the user (Petford et al., 2018). Similar to our findings, they found raycast to outperform the other techniques with reported R^2 values between 84 and 95. Their Fitts' model analysis using angular width and height outperformed the ones using absolute distances. These findings align with Jota et al. (2010) and Kopper et al. (2010) who both also used angular width and height for their Fitts' model formulation. Both Petford et al. (2018), Jota et al. (2010) and Kopper et al. (2010) used a simple Fitts' model with three parameters, while our model uses five. More parameters means more degrees-of-freedom, which can lead to over-fitting. However, the inverse is possible: a model with few degrees-of-freedom cannot capture or generalize data embedded in a high dimensional space. We believe the relative complexity of the SAR pointing task due to a complex SAR display geometry means the task cannot be captured by a simpler model. Petford et al. (2018) utilized the Shannon-Welford formulation of Fittss law to model pointing in a 3D cubemap-like projection, something that is not possible with general SAR pointing. Our model may have more in common with the trivariate pointing task modelled by Grossman and Balakrishnan (2004) who use a five parameter model.

7.4. *Design implications*

The power of predictive models, like Fitts' law, lies in their ability to simplify complex behaviour to a manageable set of measurable phenomena. Similar to other evaluations of predictive models in realistic settings (Rohs et al., 2011), we demonstrate the performance of our model across two datasets where one has more realistic target acquisition tasks and the other is more tightly controlled. All three techniques have advantages and disadvantages when pointing at targets within a SAR environment. Based on our results, we believe there are three main implications:

1. Target proximity can be used to determine technique choice. Raycast is appropriate for far targets outside the reach of the user, but for targets on or near the floor, viewport should be considered. How to switch between each pointing technique is an open question. For example, a spatial mode switching method could be used, such as recognizing when the phone is held near the body to trigger viewport pointing, or recognizing when the phone is extended away from the body for raycast pointing (Hartmann et al., 2020).
2. Occlusion can help determine target placement. It is important to evaluate the spatial configuration of the space and how it could change over time. Attaching content the user frequently interacts with in a place that has a high likelihood to become occluded will slow down selection, possibly introducing user frustration and dissatisfaction. However, there are times when occlusion could be beneficial to help deter users from undesired behaviour. For example, partially hiding a social media feed to reduce distraction. Using our model, the estimated pointing times can be calculated in real-time based on user position relative to a SAR user interface. This

means that an underlying application could adjust content placement to optimize pointing time given the context and task.

3. Direct input should only be used for targets near the user. Based on our experiment results, direct requires significantly more physical movement to reach far targets. This can increase time and cause fatigue when frequent selection is necessary. However, it may be desirable under certain circumstance, like for a fitness application or to incorporate forced movement as a means to discourage sedentary activities and to promote good health.

7.5. Limitations

We did not compare all possible pointing techniques, such as a “zoomable” viewport (Boring et al., 2009), world-in-miniature (Bowman et al., 1999), or a “perspective cursor” technique (Nacenta et al., 2006). These could be implemented on a mobile phone, but it was unclear how they could scale to a complex SAR environment. Arguably, the perspective cursor technique is an interesting candidate for investigation within a SAR environment, which could be adapted by simplifying selection to all non-occluded targets from a particular perspective. MeetAlive (Fender et al., 2017) also presented pointing techniques in a SAR environment, but similar to the work by Petford et al. (2018), the surface geometry is largely constrained to a cubemap projection which would not apply to the more complex surface geometry in our setup. Overall, all relative pointing techniques appear to make implicit assumptions about room geometry, such as using primarily large visible planes.

Recall that there could be multiple errors per trial, and that we included trials with one or more errors in movement time analysis. Another approach to analyze movement time is to use only error-free trials. To investigate whether this alternative analysis would make a difference, we used a two sample Kolmogorov-Smirnov test (Hodges, 1958) to compare all trial times with only error-free trials. There is no statistical difference ($p > 1.0 =$).

We made a decision to use a rendered camera view for the viewport condition. This is in contrast to previous evaluations of similar viewport techniques (Boring et al., 2009; Rohs and Oulasvirta, 2008; Rohs et al., 2011). Our main justification was to eliminate confounds from a potentially unreliable computer vision algorithm. Unlike those prior evaluations, we also benefited from having a high quality 3D representation of the real environment and accurate tracking.

Though there are some apparent limitations to this approach. First, the environment seen through the phone will not represent lighting, contrast, and variation in texture that a image would portray, which could limit the realistic representation of environment. However, we believe the reduction of potential confounds outweigh potential issues of this approach, and ultimately increase overall validity of the study. A modification to our approach would have been to use the 3D representation of the environment as the underlying geometric hit-detection model, but show the live video stream from the camera for user feedback. Potential pitfalls of this could be any slight misalignment of the video and 3D geometry due to otherwise imperceptible differences in spatial or temporal alignment. However, we did not consider this modified method during our process.

For Experiment 2, we use an an AR HMD to simulate a SAR pointing task. There are obvious trade-offs to this approach, such as the lack of real haptic feedback for the direct method, and side effects from wearing and using an HMD such as a limited field-of-view, weight, and mobility. Our motivation, as explained above, was to tightly control SAR pointing tasks to more deeply investigate key factors identified by the results of the the first experiment (which was conducted in a realistic SAR environment). With the ability to create and move walls and targets around a space freely inside a simulated environment, we could strictly control target occlusion, distance, and size of the targets as they relate to the actuated walls in the environment all while maintaining consistency across each participants varying head position and orientation. This allows us to test and evaluate occlusion and spatial placement precisely without the need to do something extreme, like physically actuate walls using a complex set of robot arm extensions.

8. Conclusion

In this paper, we examined fundamental characteristics of device-based interaction in SAR: pointing at surface mapped targets. Our results show how the simplicity and speed of raycasting results in excellent performance for many situations, and how surprisingly versatile a simple method like directly tapping the phone to a target can be in many situations. Our results for our implementation of the viewport pointing method is more mixed. In the ad hoc realistic SAR setting of Experiment 1, the viewport could approach raycasting performance, but was never significantly better in the tested tasks. In the controlled and more restricted setting of Experiment 2, the viewport method outperformed raycasting for distant targets that were facing the user. Our conclusion is that each method has beneficial characteristics, and that depending on the expected SAR usage context, a hybrid method or mode-switching technique to switch between methods could be the best solution. Regardless, across all techniques, we found SAR pointing characteristics like target occlusion, user movement, and target view angle to be critical factors when modelling and predicting movement time for pointing tasks. Our model incorporating these aspects was a better predictor of movement time across distant pointing techniques in both experiments, with the *SAR* ℓ_2 variation being the best generalized version. We hope our results demonstrate how empiricism and formal modelling can be applied to the new world of SAR interaction.

CRediT authorship contribution statement

Jeremy Hartmann: Conceptualization, Methodology, Formal analysis, Validation, Software, Writing - original draft. **Daniel Vogel:** Conceptualization, Validation, Writing - original draft, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests.

References

Accot, J., Zhai, S., 2003. Refining Fitts’ law models for bivariate pointing. Proceedings of the Conference on Human Factors in Computing Systems – CHI ’03. ACM Press, New York, New York, USA, p. 193. https://doi.org/10.1145/642611.642646.

Baldauf, M., Fröhlich, P., Lasinger, K., 2012. A scalable framework for markerless camera-based smartphone interaction with large public displays. 2012 International Symposium on Pervasive Displays, PerDis 2012. FTW Telecommunications Research Center Vienna, Donau-City-Strasse 1, Vienna, Austria. https://doi.org/10.1145/2307798.2307802.

Benko, H., Wilson, A.D., Zannier, F., 2014. Dyadic projected spatial augmented reality. Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST), pp. 645–655. https://doi.org/10.1145/2642918.2647402.

Bergé, L.-P., Serrano, M., Perelman, G., Dubois, E., 2014. Exploring smartphone-based interaction with overview+detail interfaces on 3d public displays. Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices &38; Services. ACM, New York, NY, USA, pp. 125–134. https://doi.org/10.1145/2628363.2628374.

Boring, S., Baur, D., Butz, A., Gustafson, S., Baudisch, P., 2010. Touch projector: mobile interaction through video. SIGCHI Conference on Human Factors in Computing Systems (CHI’10). ACM Press, New York, New York, USA, pp. 2287–2296. https://doi.org/10.1145/1753326.1753671.

Boring, S., Jurmu, M., Butz, A., 2009. Scroll, tilt or move it: using mobile phones to continuously control pointers on large public displays. pp. 161–168. doi:10.1145/1738826.1738853.

Boritz, J., Booth, K.S., 1997. A study of interactive 3D point location in a computer simulated virtual environment. Proceedings of the ACM Symposium on Virtual Reality Software and Technology – VRST ’97, pp. 181–187. https://doi.org/10.1145/261135.261168.

Bowman, D.A., Hodges, L.F., 1997. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. Proceedings of the 1997 Symposium on Interactive 3D Graphics – SI3D '97, p. 35. https://doi.org/10.1145/253284.253301.

Bowman, D.A., Johnson, D.B., Hodges, L.F., 1999. Testbed evaluation of virtual environment interaction techniques. Presence 10 (1), 26–33. https://doi.org/10.1145/323663.323667.

Box, G.E., Cox, D.R., 1964. An analysis of transformations. J. R. Stat. Soc. Ser. B 211–252.

Bragdon, A., DeLine, R., Hinckley, K., Morris, M.R., 2011. Code space: touch + air gesture hybrid interactions for supporting developer meetings. Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces – ITS '11. ACM Press, New York, USA, p. 212. https://doi.org/10.1145/2076354.2076393.

Cashion, J., Wingrave, C., Laviola, J.J., 2012. Dense and dynamic 3D selection for game-based virtual environments. IEEE Trans. Vis. Comput.Graph. 18 (4), 634–642. https://doi.org/10.1109/TVCG.2012.40.

Cashion, J., Wingrave, C., Laviola, J.J., 2013. Optimal 3D selection technique assignment using real-time contextual analysis. IEEE Symposium on 3D User Interface 2013, 3DUI 2013 – Proceedings, pp. 107–110. https://doi.org/10.1109/3DUI.2013.6550205.

Cha, Y., Myung, R., 2013. Extended Fitts' law for 3D pointing tasks using 3D target arrangements. Int. J. Ind. Ergon. 43 (4), 350–355. https://doi.org/10.1016/j.ergon.2013.05.005.

Cheng, L.-P., Chang, L., Marwecki, S., Baudisch, P., 2018. iturk: turning passive haptics into active haptics by making users reconfigure props in virtual reality. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. ACM, p. 89.

Dang, N.-T., 2007. A Survey and classification of 3D pointing techniques. 2007 IEEE International Conference on Research, Innovation and Vision for the Future. IEEE, pp. 71–80. https://doi.org/10.1109/RIVF.2007.369138.

Dolce, A., Nasman, J., Cutler, B., 2012. ARmy: a study of multi-user interaction in spatially augmented games. 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. IEEE, pp. 43–50. https://doi.org/10.1109/CVPRW.2012.6239198.

Fender, A.R., Benko, H., Wilson, A., 2017. MeetAlive: room-scale omni-directional display system for multi-user content and control sharing. Proceedings of the Interactive Surfaces and Spaces on ZZZ – ISS '17. ACM Press, New York, New York, USA, pp. 106–115. https://doi.org/10.1145/3132272.3134117.

Gervais, R., Frey, J., Hachet, M., 2015. Pointing in spatial augmented reality from 2D pointing devices. Human-Computer Interaction – INTERACT 2015. Springer International Publishing, Cham, pp. 381–389. https://doi.org/10.1007/978-3-319-22723-8_30.

Geurts, P., Ernst, D., Wehenkel, L., 2006. Extremely randomized trees. Mach. Learn. 63 (1), 3–42. https://doi.org/10.1007/s10994-006-6226-1.

Gousios, G., Pinzger, M., van Deursen, A., 2014. An exploratory study of the pull-based software development model. Proceedings of the 36th International Conference on Software Engineering - ICSE 2014. ACM Press, New York, New York, USA, pp. 345–355. https://doi.org/10.1145/2568225.2568260.

Grossman, T., Balakrishnan, R., 2004. Pointing at trivariate targets in 3D environments. Proceedings of the 2004 Conference on Human Factors in Computing Systems – CHI '04, 6, pp. 447–454. https://doi.org/10.1145/985692.985749.

Hardy, R., Rukzio, E., 2008. Touch & interact: touch-based interaction of mobile phones with displays. Proceedings of the 10th International Conference on Human Computer Interaction With Mobile Devices and Services –MobileHCI '08. ACM Press, New York, New York, USA, p. 245. https://doi.org/10.1145/1409240.1409267.

Hartmann, J., Gupta, A., Vogel, D., 2020. Extend, push, pull: smartphone mediated interaction in spatial augmented reality via intuitive mode switching. Symposium on Spatial User Interaction. ACM, New York, NY, USA, pp. 1–10. https://doi.org/10.1145/3385959.3418456.

Hartmann, J., Holz, C., Ofek, E., Wilson, A.D., 2019. RealityCheck: blending virtual environments with situated physical reality. Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19. ACM Press, New York, New York, USA, pp. 1–12. https://doi.org/10.1145/3290605.3300577.

Hartmann, J., Vogel, D., 2018. An evaluation of mobile phone pointing in spatial augmented reality. Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18. ACM Press, New York, New York, USA, pp. 1–6. https://doi.org/10.1145/3170427.3188535.

Hincapié-Ramos, J.D., Ozacar, K., Irani, P.P., Kitamura, Y., 2015. GyroWand: IMU-based raycasting for augmented reality head-mounted displays. Proceedings of the 3rd ACM Symposium on Spatial User Interaction - SUI '15. ACM Press, New York, New York, USA, pp. 89–98. https://doi.org/10.1145/2788940.2788947.

Hodges, J.L., 1958. The significance probability of the Smirnov two-sample test. Arkiv för Matematik 3 (5), 469–486.

ISO/TS 9241-411, 2012. Ergonomics of human-system interaction – Part 411: Evaluation methods for the design of physical input devices. Standard. International Organization for Standardization. Geneva, CH.

Jones, B., Sodhi, R., Murdock, M., Mehra, R., Benko, H., Wilson, A., Ofek, E., MacIntyre, B., Raghuvanshi, N., Shapira, L., 2014. Roomalive: magical experiences enabled by scalable, adaptive projector-camera units. Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology. Association for Computing Machinery, New York, NY, USA, pp. 637–644. https://doi.org/10.1145/2642918.2647383.

Jones, B.R., Benko, H., Ofek, E., Wilson, A.D., 2013. Illumiroom: peripheral projected illusions for interactive experiences. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, pp. 869–878. https://doi.org/10.1145/2470654.2466112.

Joshi, N., Vogel, D., 2019. An evaluation of touch input at the edge of a table. Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, pp. 1–12. https://doi.org/10.1145/3290605.3300476.

Jota, R., Nacenta, M.A., Jorge, J.A., Carpendale, S., Greenberg, S., 2010. A comparison of ray pointing techniques for very large displays. Proceedings of Graphics Interface 2010. Canadian Information Processing Society, CAN, pp. 269–276.

Kopper, R., Bowman, D.A., Silva, M.G., McMahan, R.P., 2010. A human motor behavior model for distal pointing tasks. Int. J. Hum. Comput. Stud. 68 (10), 603–615. https://doi.org/10.1016/j.ijhcs.2010.05.001.

Lopes, P., Jota, R., Jorge, J.A., 2011. Augmenting touch interaction through acoustic sensing. Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces - ITS '11. ACM Press, New York, New York, USA, p. 53. https://doi.org/10.1145/2076354.2076364.

Machuca, M.D.B., Chinthammit, W., Yang, Y., Duh, H., 2014. 3D mobile interactions for public displays. SIGGRAPH Asia 2014 Mobile Graphics and Interactive Applications, SA 2014. Association for Computing Machinery, Inc, Human Interface Technology Laboratory Australia, University of Tasmania, Australia. https://doi.org/10.1145/2669062.2669074.

MacKenzie, I.S., 1992. Fitts' law as a research and design tool in human-computer interaction. Hum.–Comput. Interact. 7 (1), 91–139. https://doi.org/10.1207/s15327051hci0701_3.

MacKenzie, I.S., Buxton, W., 1992. Extending Fitts' law to two-dimensional tasks. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems – CHI '92. ACM Press, New York, New York, USA, pp. 219–226. https://doi.org/10.1145/142750.142794.

Marquardt, N., Diaz-Marino, R., Boring, S., Greenberg, S., 2011. The proximity toolkit: prototyping proxemic interactions in ubiquitous computing ecologies. Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology. Association for Computing Machinery, New York, NY, USA, pp. 315–326. https://doi.org/10.1145/2047196.2047238.

Meyer, D.E., Abrams, R.A., Kornblum, S., Wright, C.E., Smith, J.E., 1988. Optimality in human motor performance: ideal control of rapid aimed movements. Psychol. Rev. 95 (3), 340–370. https://doi.org/10.1037/0033-295X.95.3.340.

Molyneaux, D., Izadi, S., Kim, D., Hilliges, O., Hodges, S., Cao, X., Butler, A., Gellersen, H., 2012. Interactive environment-aware handheld projectors for pervasive computing spaces. Proceedings of the 10th International Conference on Pervasive Computing. Springer-Verlag, Newcastle, UK, pp. 197–215. https://doi.org/10.1007/978-3-642-31205-2_13.

Mur-Artal, R., Tardos, J. D., 2016. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. arXiv (October). 1610.06475.

Murata, A., Iwase, H., 2001. Extending Fitts' law to a three-dimensional pointing task. Hum. Mov. Sci. 20, 791–805.

Myers, B.A., Peck, C.H., Nichols, J., Kong, D., Miller, R., 2001. Interacting at a distance using semantic snarfing. In: Abowd, G.D., Brumitt, B., Shafer, S. (Eds.), Ubicomp 2001: Ubiquitous Computing. Springer, Berlin, Heidelberg, pp. 305–314. https://doi.org/10.1007/3-540-45427-6_26.

Nacenta, M.A., Aliakseyeu, D., Subramanian, S., Gutwin, C., 2005. A comparison of techniques for multi-display reaching. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems – CHI '05. ACM Press, New York, New York, USA, p. 371. https://doi.org/10.1145/1054972.1055024.

Nacenta, M.A., Gutwin, C., Aliakseyeu, D., Subramanian, S., 2009. There and back again: cross-display object movement in multi-display environments. Hum.–Comput. Interact. 24 (1–2), 170–229. https://doi.org/10.1080/07370020902819882.

Nacenta, M.a., Sallam, S., Champoux, B., Subramanian, S., Gutwin, C., 2006. Perspective cursor: perspective-based interaction for multi-display environments. Proceedings of the SIGCHI conference on Human Factors in computing systems, 15, p. 298. https://doi.org/10.1145/1124772.1124817.

Nancel, M., Pietriga, E., Chapuis, O., Beaudouin-Lafon, M., 2015. Mid-air pointing on ultra-walls. ACM Trans. Comput.-Hum. Interact. 22 (5), 21:1–21:62. https://doi.org/10.1145/2766448.

Ondruska, P., Kohli, P., Izadi, S., 2015. MobileFusion: real-time volumetric surface reconstruction and dense tracking on mobile phones. IEEE Trans. Vis. Comput. Graph. 21 (11), 1251–1258. https://doi.org/10.1109/TVCG.2015.2459902.

Oswald, P., Tost, J., Wettach, R., 2014. The real augmented reality. Proceedings of the 11th Conference on Advances in Computer Entertainment Technology - ACE '14. ACM Press, New York, New York, USA, pp. 1–4. https://doi.org/10.1145/2663806.2663853.

Parker, J.K., Mandryk, R.L., Inkpen, K.M., 2005. Tractorbeam: seamless integration of local and remote pointing for tabletop displays. Proceedings of Graphics Interface 2005. Canadian Human-Computer Communications Society, Waterloo, CAN, pp. 33–40.

Pejsa, T., Kantor, J., Benko, H., Ofek, E., Wilson, A.D., 2016. Room2Room: enabling life-size telepresence in a projected augmented reality environment. Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing - CSCW '16. ACM Press, New York, New York, USA, pp. 1714–1723. https://doi.org/10.1145/2818048.2819965.

Petford, J., Nacenta, M.A., Gutwin, C., 2018. Pointing all around you: selection performance of mouse and ray-cast pointing in full-coverage displays. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. ACM, New York, NY, USA, pp. 533:1–533:14. https://doi.org/10.1145/3173574.3174107.

Poupyrev, I., Billinghurst, M., Weghorst, S., Ichikawa, T., 1996. The go-go interaction technique: non-linear mapping for direct manipulation in VR. Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology. Association for Computing Machinery, New York, NY, USA, pp. 79–80. https://doi.org/10.1145/237091.237102.

Raskar, R., Welch, G., Cutts, M., Lake, A., Stesin, L., Fuchs, H., 1998. The Office of the Future: a Unified Approach to Image-based Modeling and Spatially Immersive

Displays. ACM, New York, NY, USA, pp. 179–188. https://doi.org/10.1145/280814.280861.

Raskar, R., Welch, G., Fuchs, H., 1999. Spatially augmented reality. Proceedings of the International Workshop on Augmented Reality: Placing Artificial Objects in Real Scenes: Placing Artificial Objects in Real Scenes. A. K. Peters, Ltd., USA, pp. 63–72.

Raskar, R., Welch, G., Low, K.-L., Bandyopadhyay, D., 2001. Shader lamps: animating real objects with image-based illumination. Proceedings of the 12th Eurographics Workshop on Rendering Techniques, pp. 89–102. https://doi.org/10.1007/978-3-7091-6242-2_9.

Ray, B., Posnett, D., Filkov, V., Devanbu, P.T., 2014. A large scale study of programming languages and code quality in Github categories and subject descriptors. SIGSOFT International Symposium on Foundations of Software Engineering, pp. 155–165.

Rekimoto, J., 1997. Pick-and-drop: a direct manipulation technique for multiple computer environments. Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology. Association for Computing Machinery, New York, NY, USA, pp. 31–39. https://doi.org/10.1145/263407.263505.

Rohs, M., Oulasvirta, A., 2008. Target acquisition with camera phones when used as magic lenses. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, pp. 1409–1418. https://doi.org/10.1145/1357054.1357275.

Rohs, M., Oulasvirta, A., Suomalainen, T., 2011. Interaction with magic lenses: real-world validation of a Fitts' law model. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, pp. 2725–2728. https://doi.org/10.1145/1978942.1979343.

Schmidt, D., Chehimi, F., Rukzio, E., Gellersen, H., 2010. Phonetouch: a technique for direct phone interaction on surfaces. Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology. Association for Computing Machinery, New York, NY, USA, pp. 13–16. https://doi.org/10.1145/1866029.1866034.

Schmidt, D., Seifert, J., Rukzio, E., Gellersen, H., 2012. A cross-device interaction style for mobiles and surfaces. Proceedings of the Designing Interactive Systems Conference. Association for Computing Machinery, New York, NY, USA, pp. 318–327. https://doi.org/10.1145/2317956.2318005.

Seifert, J., Bayer, A., Rukzio, E., 2013. Pointerphone: using mobile phones for direct pointing interactions with remote displays. Human-Computer Interaction – INTERACT 2013. Springer Berlin, pp. 18–35. https://doi.org/10.1007/978-3-642-40477-1_2.

Seifert, J., Schneider, D., Rukzio, E., 2013. Extending mobile interfaces with external screens. Human-Computer Interaction – INTERACT 2013. Springer Berlin, pp. 722–729. https://doi.org/10.1007/978-3-642-40480-1_50.

She, J., Crowcroft, J., Fu, H., Ho, P.-H., 2013. Smart signage: An interactive signage system with multiple displays. 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, pp. 737–742. https://doi.org/10.1109/GreenCom-iThings-CPSCom.2013.133.

Siek, K.A., Rogers, Y., Connelly, K.H., 2005. Fat finger worries: how older and younger users physically interact with PDAs. Human-Computer Interaction - INTERACT 2005. Springer, Berlin, Heidelberg, pp. 267–280. https://doi.org/10.1007/11555261_24.

Teather, R.J., Stuerzlinger, W., 2010. Target pointing in 3D user interfaces. CEUR Workshop Proc. 588, 20–21.

Teather, R.J., Stuerzlinger, W., 2011. Pointing at 3D targets in a stereo head-tracked virtual environment. 2011 IEEE Symposium on 3D User Interfaces (3DUI). IEEE, pp. 87–94. https://doi.org/10.1109/3DUI.2011.5759222.

Teather, R.J., Stuerzlinger, W., 2013. Pointing at 3D target projections with one-eyed and stereo cursors. Conference on Human Factors in Computing Systems – Proceedings, pp. 159–168. https://doi.org/10.1145/2470654.2470677.

Wilson, A., Shafer, S., 2003. Xwand: Ui for intelligent spaces. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, pp. 545–552. https://doi.org/10.1145/642611.642706.

Wobbrock, J.O., Findlater, L., Gergle, D., Higgins, J.J., 2011. The aligned rank transform for nonparametric factorial analyses using only anova procedures. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, pp. 143–146. https://doi.org/10.1145/1978942.1978963.

Jeremy Hartmann is a PhD candidate at the School of Computer Science at the University of Waterloo. His research interests focus on virtual and augmented reality within Human Computer Interaction.

Daniel Vogel is an Associate Professor in the School of Computer Science at the University of Waterloo. His research area is Human Computer Interaction, focusing on fundamental characteristics of human input and novel forms of interaction for current and future computing form factors.