Evaluating Menu Techniques for Handheld AR with a Smartphone & Mid-Air Pen

Philipp Wacker
wacker@cs.rwth-aachen.de
RWTH Aachen University
Aachen, Germany

Oliver Nowak
nowak@cs.rwth-aachen.de
RWTH Aachen University
Aachen, Germany

Simon Voelker
voelker@cs.rwth-aachen.de
RWTH Aachen University
Aachen, Germany

Jan Borchers
borchers@cs.rwth-aachen.de
RWTH Aachen University
Aachen, Germany

Figure 1: Handheld AR systems with a mid-air pointing device let users interact in Augmented Reality requiring only a smartphone and a custom printed pen (left). We implemented and evaluated menu techniques that use different methods for the interaction with the system, for example, using the mid-air pen (middle) or the hand holding the smartphone (right).

ABSTRACT

Adding a mid-air pen to Handheld Augmented Reality creates a new kind of bimanual interaction for which many fundamental interaction design questions have not been answered yet. In particular, menus are an essential component in most visual interfaces, but it is unclear how to best interact with them in this setting: using the pen in mid-air or on a surface, using the touchscreen, or by moving the smartphone itself. We compared basic menus for these methods by analyzing success rates, selection times, device movement, and subjective ratings. Our results indicate that interacting with a mid-air menu using the pen, and operating a menu with the hand holding the smartphone, are sufficiently competitive to the current standard of two-handed touchscreen interaction, so that interaction designers can freely choose among them based on the interaction context of their application.

CCS CONCEPTS

• **Human-centered computing → Human computer interaction (HCI)**; *Mixed / augmented reality*; *Interaction techniques.*

KEYWORDS

Menu techniques; Augmented Reality; Handheld; Mobile; Smartphone; AR; Bimanual; Mid-air; Interaction; 3D; ARPen

ACM Reference Format:

Philipp Wacker, Oliver Nowak, Simon Voelker, and Jan Borchers. 2020. Evaluating Menu Techniques for Handheld AR with a Smartphone & Mid-Air Pen. In *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '20), October 5–8, 2020, Oldenburg, Germany*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3379503.3403548

MobileHCI '20, October 5–8, 2020, Oldenburg, Germany

ACM ISBN 978-1-4503-7516-0/20/10…$15.00
https://doi.org/10.1145/3379503.3403548

1 INTRODUCTION

Mid-air interaction for Augmented Reality (AR) environments is a very active field of research and Handheld AR, such as on smartphones, has become more widespread due to the introduction of AR frameworks into the major operating systems[1]. However, it is hard to specify a point in mid-air while only using a touchscreen since the depth dimension is missing when touching a 2D position on the screen. In a previous work, we developed the ARPen system that combines a smartphone based AR application with a mid-air pointing device which is tracked by the smartphone [30]. Since no additional tracking equipment is needed, this system presents a

[1] developer.apple.com/arkit, developers.google.com/ar (accessed June 12, 2020)

very quick & easy way to enable accurate 3D input in a mobile AR scenario. This can, for example, be useful for Personal Fabrication for which objects could be designed in Augmented Reality *within* the real environment before printing the object using a 3D printer. A user could take the ARPen into her car and design an inset for the cup holder to secure smaller sized cans or she could design fitting attachments for flower pots on her balcony. However, many aspects of the interaction with such a system have not been evaluated yet.

One of the central interaction tasks is *system (*or *application) control* [4, 8]. An example of these tasks are menu interactions. Menus are an essential component in most visual interfaces and they can take many different forms. One kind of menu is the context menu [8] which is often used to adjust the properties of a specific object. In the context of an Augmented Reality modeling application, such properties could, for example, be the color or transparency of an object in the scene. We asked ourselves the question how context menus can be realized for a Handheld AR system with a mid-air pointing device. Such a system offers different interaction methods that could be used to interact with the menu. Besides using the touchscreen, the mid-air pointing device can be used for interaction. Also, the device position and orientation in space might be beneficial to use as it would not require to move the mid-air pointing device. Since the physical world can improve the interaction in mid-air systems [1, 31], a menu on a physical surface in the scene might also improve menu interaction. The different interaction methods can all have different advantages and disadvantages for the interaction with menus in an AR application.

We implemented basic menus for the interaction methods and evaluated them in a user study. We compared their performance and collected subjective preferences of users. Based on the results of our study, we consider advantages and disadvantages of each interaction method to assist interaction designers in picking suitable menus for different contexts.

In summary, our contributions are:

- an empirical study about menus using different interaction methods of Handheld AR systems;
- design considerations for such menus based on their advantages and disadvantages.

In the next section, we provide an overview of the related work in the area of mid-air Handheld AR interaction as well as previous research regarding menu interaction with a focus on VR and AR environments. Afterwards, we explain the different menus we used in our study and present the study design and procedure, before discussing the results.

2 RELATED WORK

The work presented in this paper is at the intersection between the areas *Mid-air Interaction in Handheld AR* and *Menus in VR & AR*. Methods to control an application, for example by specifying actions through menus, are one of the essential interaction tasks for immersive systems [4, 8]. New interaction methods enable new techniques to realize menus and require further exploration.

2.1 Mid-Air Interaction in Handheld AR

Most current Handheld AR systems use input on the touchscreen to interact with the virtual content, for example, to select and move items (e.g., [25, 27, 29]) or to sketch and model 3D objects by projecting touch drawings into the scene—either onto previously defined drawing planes [34] or onto tracked surfaces in the world [16]. However, specifying precise points in mid-air—such as to allow freehand sketching and drawing—is problematic since the input on the touchscreen is missing the 3rd dimension. Researchers have, for example, added depth cameras to their setup to track input in mid-air (e.g., [2, 3]).

Using a tracking similar to the DodecaPen [33], we previously developed the ARPen system which combines the surface tracking technology in recent smartphones with marker tracking to track a custom build pen in mid-air [30]. Apart from the markers used for tracking, this pen contains three buttons and their state is transmitted to the smartphone via Bluetooth. With this pen, we evaluated different techniques of selecting and moving objects in mid-air, and found that raycasting approaches achieve the best results for selection tasks. The ray could either be cast from a touch on the screen of the phone or cast from the camera through the tip of the pen (selecting the element behind the pen), which we called *pen-ray selection*. This raycasting style of selecting objects works well in Handheld AR due to the 2D projection of the 3D world on the screen. However, a similar technique in stereoscopic systems would require additional adjustments since no singular projection plane exists. On the other hand, the 2D projection also increases issues of perceiving the correct depth of a virtual object, which is a well known issue in AR (see [21] for a survey). Recently, we evaluated different techniques to visualize the distance between the input device and objects in the scene to address this issue [32].

For most of these projects, users interact with the system using one interaction method, and the focus of studies is mostly on *selection* and *manipulation* rather than *system control* tasks. So far no study has investigated the effect of different interaction methods on the performance of *system control* tasks in the form of menus. In our study, we used a Handheld AR system with a mid-air pen to evaluate the performance of basic menus using the different interaction methods.

There are many use cases in which it is necessary to specify a precise mid-air position. Throughout this paper, we use the example of a mid-air modeling system as described in the introduction as our usage scenario.

2.2 Menus in VR & AR

Early HCI studies suggest that pie menus are faster compared to linear menus [5, 10]. With regard to the number of elements in a menu, studies have found that eight items per level achieves good results and this number is used in several menu evaluations [18, 22, 24].

Dachselt and Hübner [8] present a good overview and a taxonomy for menus in virtual environments. The central elements that they use to classify menus are *Intention of use* (e.g., number of items), *Appearance and Structure* (e.g., structural layout), *Placement* (e.g., reference), *Invocation and Availability* (e.g., visibility), *Interaction and I/O setting* (e.g., interaction device), *Usability* (e.g., evaluation criteria), and *Combinability*. The menu techniques evaluated in this work occupy similar spaces in this taxonomy but differ on aspects defined by the different interaction methods.

Menus in VR and AR have been the subject of research for a long time [13, 15, 28]. One of the earliest projects is by Jacoby and Ellis [15], who placed linear menus as floating windows in a VR environment and interacted with them via hand-pointing gestures. Feiner et al. [13] were one of the first to implement menus for AR contexts. They discern the placement and attachment of menus in the scene between *surround-fixed*, *display-fixed*, and *world-fixed* windows. This differentiation has also been picked up in other projects (e.g., [8, 19, 23]).

For Virtual Reality, Das and Borst [10] evaluated various menu setups and interactions. They compared ray-casting to techniques that attach a cursor to the menu and control it indirectly (PAM: pointer-attached-to-menu). Furthermore, they compared contextual menus around a specific object against menus that are fixed in the world. Their results indicate that ray-casting control is generally faster compared to indirect setups and users prefer ray-casting techniques. The contextual menus around a specific object were used faster than fixed-world menus. However, the authors did not detect differences in error rates between these menu styles [10]. Dang and Mestre [9] found out that the performance of menus also depends on their orientation in space and that accuracy on horizontal menus decreases compared to more angled menus. Lee et al. [23] developed a system for head-mounted AR in which a smartphone is used as a controller to interact with menus. However, they did not perform a user study to compare the different interactions.

In our paper, we present and evaluate different menu techniques for a Handheld AR system with a mid-air pen. Such a system presents multiple methods that users can use to interact with the system and each might provide different advantages and disadvantages for *system control* tasks such as menus.

3 MENU TECHNIQUES

In this section, we describe the menu techniques we designed and implemented for our study. We based our designs on the main *methods of interaction* that are directly available to a Handheld AR system with a mid-air pointing device since they are the necessary hard- & software capabilities for such a system to work.

The first *method* is the interaction with the touchscreen of the smartphone. The screen is needed to show virtual objects in the scene and enable the user to interact with them. To interact with virtual objects, a ray is cast into the scene to decide which object is being targeted (e.g., [25, 26]). The touchscreen can also be used to show static elements at fixed positions on the screen. The mid-air pen is another essential part of such systems. Therefore, the next *method* is the interaction with this pen. This interaction can either be mid-air—used, for example in sketching applications to draw lines in mid-air—or by tracing over physical surfaces in the scene. These surfaces have to be tracked by the system to calculate the position of the phone in the scene [30]. We distinguish between these methods that use the pen, since interacting with physical surfaces can aid the precision of input in immersive environments [1, 31] but it is restricted in its availability while mid-air interaction can be in-place all the time. The third *method* to interact with the system is through the movement of the viewing device itself. Adjusting the viewport into the virtual world is an essential part of Handheld AR and the changes of orientation and position of the device can be used as input parameters. Other methods of interaction, such as back-of-device interaction or voice interaction, add additional layers to the interaction with a Handheld AR system with a mid-air pen since they are not required for the main functionality of the system. Therefore, we focused on the main interaction methods that such a system offers directly.

Table 1: Combination of object and item selection technique for each menu technique

Menu Technique	Object Selection	Item Selection
Two-handed touch	Two-Handed	Two-Handed
Mid-air pen	Pen-Ray	Pen-Ray
One-handed touch	Pen-Ray	One-Handed
Device pointer	Pen-Ray	Device Motion
Surface	Pen-Ray	Surface Tap

Interacting with a menu requires at least two steps. The first step is to open the menu. For a context menu, this means that the user has to select the object to open the menu for (*object selection*). The second step is to select an item from the menu (*item selection*).

Since opening the menu is a standard object selection task, we used the two most promising techniques from our previous study [30] for this step. In that study, the raycasting techniques *two-handed touch selection* and *pen-ray selection* performed with similar success and preference. For *two-handed touch selection* the ray is cast from the location of the touch on the screen and for *pen-ray selection*, the ray is shot from the camera through the pen tip into the scene. The first target hit is returned as the selection. For the *object selection* step of our menu techniques, we picked a selection technique based on the main interaction method and our use case scenario of mid-air modeling. Only for our baseline condition of the two-handed touch menu, we decided to use the *two-handed touch selection* technique. Since an application can require successive menu interactions before returning to mid-air input, for example setting colors for different objects, we did not want to put the two-handed touch menu at a disadvantage by using an *object selection* technique that requires additional movement instead of a technique that performs similarly well. The remaining menu techniques either use the pen for interaction, making the *pen-ray selection* technique the preferred choice, or have no clear choice of *object selection* technique, in which case *pen-ray selection* allows to hold a 3D position while interacting with the menu.

The second step of selecting an item from the menu is different from selecting an object in that it is not necessarily tied to the location of the target anymore. This enables the possibility to present the menu in different locations and also to interact with it using any of the different interaction methods, which might lead to different results compared to our previous mid-air selection study [30]. Table 1 shows the combinations of *object selection* and *item selection* for each menu technique.

We decided to use radial menus for all our mid-air menus as they have shown best results in the literature (e.g., [5, 10, 20]) and keeping them the same reduces the impact of different presentation techniques. In the center of the menu, we placed a 'back' button that allows to close the menu.

In the following, we describe the five menu techniques we used in our study.

3.1 Two-Handed Touch Menu

The two handed touch menu (*two-handed touch*, Fig. 2 a) is modeled after the most often used way of interacting with Augmented Reality content. The phone is held in the non-dominant hand while the dominant hand interacts on the screen. The menu is opened by tapping on the rendering of the virtual object on the screen. A radial menu with roughly 90% of the height of the screen then opens around the selected target in mid-air and the user can select an item by tapping on it. Once the finger touches the screen, the item underneath it is highlighted. The menu always faces the camera when the phone is moved around the object. Tapping outside of the menu or on the inner area closes the menu without a selection.

3.2 Mid-Air Pen Menu

The mid-air pen menu (*mid-air pen*, Fig. 2 b) uses the same visualization as the two-handed touch menu. However, opening the menu and selecting an item is done with the pen-ray technique. While the menu is open, only the tip of the pen is visualized in front of the menu. This allows to use the pen tip as a cursor while avoiding occlusion of items by the user's hand. Hovering over an item highlights it.

For this and the remaining menu techniques, the action after pressing a button on the pen while the menu is open depends on the element behind the visible cursor. If the cursor is over a menu item, this item is selected. If the cursor is beside the menu, the menu closes, which is a similar behavior to clicking outside of an open menu in current operating systems.

3.3 One-Handed Touch Menu

The one-handed touch menu (*one-handed touch*, Fig. 2 c) is similar to a normal table view that is shown on the left side of the touchscreen. After the menu is triggered by the mid-air selection, it is opened to the left of the touchscreen so that an item can be selected with the hand holding the phone. We chose the size of the menu based on the results of our grasp-study in [30]. The shape of the menu follows the rotational movement of the thumb making it easy to touch each element. The top space of the menu is used for the 'back' button and pressing any button on the mid-air pen closes it as well.

3.4 Device Pointer

The device pointer (*device pointer*, Fig. 2 d) uses the movement of the device to select an item in the menu. The menu is opened with a pen-ray selection and once the menu is open, a cursor is shown in the center of the menu. This cursor behaves as if a ray is cast from the camera onto the menu, which means that the translation and rotation of the smartphone is applied to the cursor. The cursor is also clamped to the size of the menu so that no overshoots are possible. If the user moves the device further than the edge of the menu, she has to return to the edge before the cursor moves again. This corresponds to an absolute mapping of the device movement onto the cursor movement. The item underneath the cursor is highlighted and pressing a button on the pen selects the item. While the cursor of this technique is shown, the sphere on the tip of the pen is removed to avoid confusion.

3.5 Surface Menu

The surface menu (*surface*, Fig. 2 e) takes the knowledge about the environment into account. After a menu is triggered by a pen-ray selection, the pie menu is shown on the table surface in front of the user. Pie menus have also shown good results in tabletop scenarios [20], so we decided upon this structure for the menu to keep the differences in visualization minimal compared to the other menu techniques. As the user moves the pen close to the menu on the table, the object she is hovering over gets highlighted. Tapping the surface and lifting the pen selects the tapped item. Since depth perception is a known problem of Handheld AR [21], we display the shadow of the pen tip on the menu to aid the user in determining over which item they are currently hovering. The other techniques work by using raycasting techniques, so that showing a shadow is not necessary for them. Due to the perspective, a menu with the same absolute size to the mid-air menu would look smaller on the screen. Therefore, we increased the size so that the rendered menu also takes up around 90% of the device height when looking from the participant's seated position. The location of the menu stays the same throughout the session so that users can remember the location even if the table is not visible through the viewport when opening the menu. World-fixed menus did not perform as well as object-fixed menus in [10] and this design likely requires longer movement times since the menu is not automatically placed close to the object selected or at the fingertips of the user. Also, this is the only technique requiring a depth estimation to move to the intended location, potentially increasing the interaction times. However, the added haptic component of the surface might improve the interaction so much that users accept the longer movement times.

While there are many other ways how menus could be implemented for the different interaction methods, the techniques we have chosen represent basic methods for each. Since the basic parameters for each menu are the same, this allows to compare the effect of each interaction method on menu performance. This also becomes apparent when placing our menu techniques into the taxonomy by Dachselt and Hübner [8], as the major differences are specific to the interaction method:

- *structural layout*: *one-handed touch* uses a list in an arc; all others use a radial pie menu;
- *reference* and *orientation*: 'virtual object' and 'user facing' for *two-handed touch*, *mid-air pen*, and *device pointer*; 'device' and 'on the side of the hand holding it' for *one-handed touch*; 'world' and 'table surface' for *surface*;
- *interaction device*: pen for *mid-air pen* and *surface*; pen + touchscreen for *one-handed touch*; touchscreen for *two-handed touch*; pen + device for *device pointer*

In the following section, we describe our study comparing the different menu techniques.

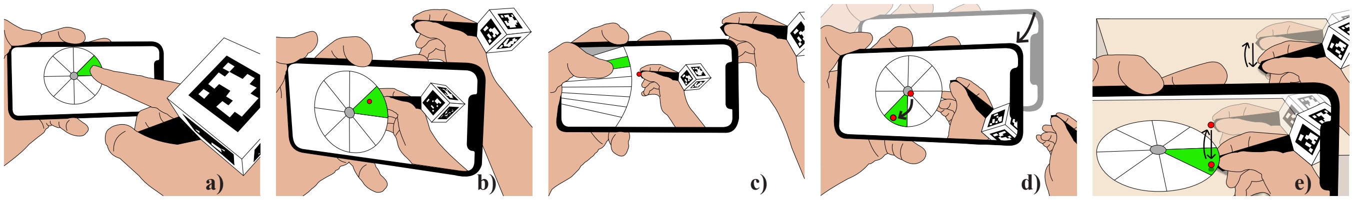

Figure 2: The menu techniques used in our study. a) *two-handed touch* as the baseline, b) *mid-air pen*, c) *one-handed touch*, d) *device pointer*, e) *surface*.

4 COMPARING MENU TECHNIQUES

To understand how the different interaction methods affect menu performance, we performed a study comparing the menu techniques in terms of successful selections, time for selection and movement of the device as well as subjective feedback regarding ease of use and comfort while using the menu. We included measurements on device movement since more movement could increase muscle strain making it harder to keep the device in hand. Particularly for *device pointer* we also wanted to see whether users preferred to use translation or rotation movement to select an item.

The task for the participants was to select an emoji shown on the screen from a menu around a virtual cube. This represents the task in which a user opens a menu with the intended option in mind but not knowing where it will be located in the menu. Keeping everything aside from the interaction method for the item selection similar between each condition enables us to judge the impact of the different interaction methods. Our results can then be used as a starting point for an interaction designer to decide which menu technique to use.

4.1 Study Setup

We implemented the menu techniques starting from our open-source ARPen system [30][2]. For our study, 16 cubes with an edge length of 3 cm were placed about 40 cm in front of the seated participant. The different cubes served as targets to open the menus for and we varied their position to avoid confounding effects from target placement. Since all techniques used raycasting methods for the selection of targets, we did not vary the depth of target objects in the scene. A marker pattern on the table ensured that the position of the cubes stayed consistent between participants. For each trial, one of the cubes was highlighted in green, marking it as the target cube the menu should be opened for. Each cube was used twice as the target cube resulting in 32 trials per menu technique. The order of the target cubes was randomized.

We used emojis as the items in the menus since they offer a strong visual and semantic differentiation suitable for our general task. The target emoji was shown in the top right corner of the screen. This emoji was randomly picked from one of eight categories: Faces, people, activities, flags, animals, food, travel, or objects. We chose the candidates for every emoji category so that they were easily distinguishable from each other. Each menu contained eight items based on the findings from related work [18, 24]. Random selections from the other emoji categories were used to fill the remaining places in the menu. The placement of the emojis inside the menu was randomized but we ensured that the target emoji was placed at every position four times for each menu technique. Selecting an emoji closed the menu and showed the next target cube and target emoji regardless of whether the selection was correct or not. Opening the menu for a cube other than the current target cube showed a selection of emojis without the target emoji.

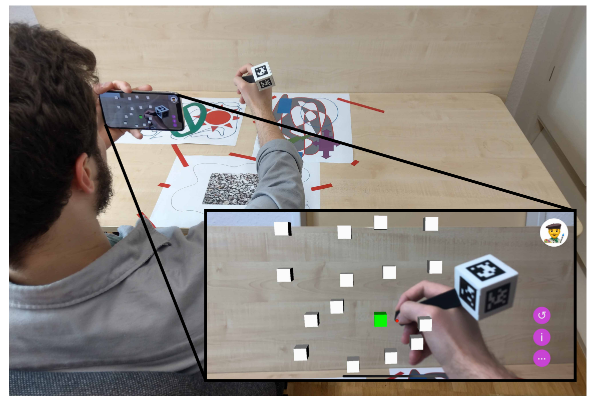

Figure 3: Participants had to open the menu for the green cube and select the emoji that was shown on the top right of the screen. The marker on the table ensured that the cube position was consistent between participants.

4.2 Study Procedure

Participants sat in front of the table with the marker patterns. Each participant was asked to hold the phone in her non-dominant hand using her index finger and pinkie, leaving the thumb available for interaction with the touchscreen (*pinkie* grasp from [30], see Fig. 1). The participant was given time to test out the grip and the interaction before we explained the menu techniques and task of the study. Since our example usage scenario of mid-air modeling requires frequent specifications of mid-air points, we asked participants to keep the pen in their hand even for the *two-handed touch* technique which does not require the use of the pen otherwise. For each technique, we explained how the menu could be opened, described the visualization, and showed how an item can be selected. The participant could then try out the menu to familiarize herself with the interaction. Once the participant was confident in using the menu, she could select a button beside the marker to switch to the recording stage. The recording stage started with one additional test trial that was not recorded. This way, all recorded trials followed the same interaction sequence. If the participant needed a rest or

[2] https://github.com/i10/ARPen (accessed June 12, 2020)

stopped to comment on the interaction, we marked the current trial as an outlier and restarted the last trial. After each cube was used twice as the target for the menu, we asked the participant to rate how easy she thought the use of the menu was and how comfortable she felt during the interaction. Also, we asked how she rated the combination of interaction techniques to open the menu and select an item as the interaction was not the same for every menu technique. During the interaction with one menu technique, the moderator noted qualitative comments about the interaction. After having used all five menu techniques, we asked participants to rank them from best to worst. In total, each participant selected 160 emojis (32 trials×5 menu techniques). The order of menu techniques was counterbalanced using a Latin-square.

4.3 Measurements

We recorded the success of every trial by determining if the correct emoji was selected. Furthermore, we recorded the time for each trial separated into the time to select the target cube and the time from showing the menu to selecting the item. For the device movement we recorded the change of movement between frames for translation and rotation individually per axis with a frequency of 30Hz. The individual measurements per axis were summed up for the overall movement indicator.

For the subjective ratings for ease of use (*ease*), comfort of the interaction (*comfort*), and combination of selection techniques (*CoST*), we recorded 7-point Likert-Scale ratings for each technique as well as a ranking of all five techniques.

4.4 Evaluation

From our initial 15 participants, we excluded the data of one user as she was the only user who repeatedly selected the wrong target cube and then selected a wrong item. Also, she opened and closed menus more often than other users. We recruited another participant to fill the Latin-square and have measurements for 15 participants (5 female, 22–54 years, M:29 years, SD:10.3 years, all right handed). Due to a recording error, 2 trials were not recorded. Overall, we collected measurements for 2398 item selections. For every participant, we summed up the number of successful selections for each menu technique and divided it by the number of repetitions to calculate the percentage of correct selections (*success*). We averaged the time to open the menu (*timeToMenu*), the time to select an item (*timeToItem*) as well as the translation and rotation movement (*translation*, *rotation*). We present our results based on the Fair Statistical Communication Guidelines by Dragicevic [11] and the Transparent Statistics Working Group [17]. Following these guidelines, we do not calculate p-values or make dichotomous decisions on significance or non-significance, but rather communicate the uncertainty in the evaluation by using the principles of *estimation* [7]. We present the effects found in our evaluation with 95% confidence intervals. For each measurement of a menu technique, we subtracted the corresponding value of the baseline value (*two-handed touch*). This way, these results show the differences between the menu techniques and the baseline condition. We bootstrapped these differences with 1000 repetitions and calculated the 95% confidence intervals using the BCa method [6, 12]. For the following graphs, this means that values to the right of zero indicate that the baseline technique had less of the measured variable, while values to the left indicate higher values for the baseline compared with the other technique. Since the interpretation of differences in Likert-scale data is not defined, we report the means and bootstrapped confidence intervals for the subjective ratings without computing the difference to the baseline.

We provide both our software and data alongside results from a more common analysis as supplementary materials[3].

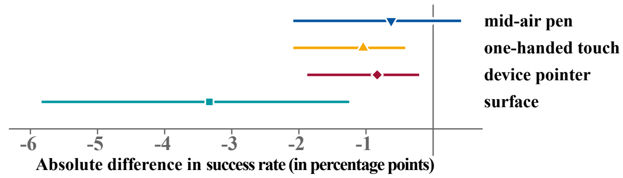

Figure 4: Differences in the success rates between the menu techniques and the baseline (M: 99.79 %; CI [99.17 %, 100 %]). Only *surface* appears to be less precise. Whiskers denote the 95% CI (confidence interval).

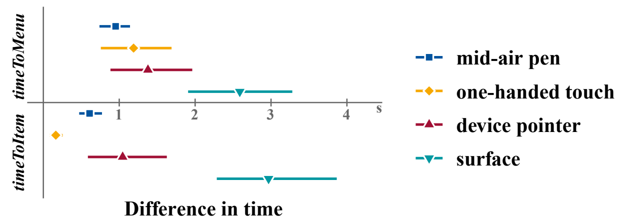

Figure 5: Difference in *timeToMenu* (top) and *timeToItem* (bottom). *Mid-air pen, one-handed touch,* and *device pointer* take 1–2s longer than the baseline to open the menu. For the item selection, *one-handed touch* is close to the performance of the baseline followed by *mid-air pen* and *device pointer*. Whiskers denote the 95% CI.

4.5 Results

In the following, we present the results of the study, first focusing on the measured data before reporting the subjective ratings of the participants and the qualitative remarks during the study.

4.5.1 Success, Time, and Device Movement. All techniques reached a high success rate (*two-handed touch* M: 99.79 %; CI [99.17 %, 100 %]) and there seem to be no differences between them with an indication that *surface* could be less successful (Fig. 4).

For *timeToMenu*, the baseline condition (*two-handed touch*, M: 1.38 s, CI [1.3, 1.5]) showed faster results compared to the other techniques (Fig. 5, top). Using the other techniques, participants had to use the pen-ray selection to open the menu. The difference of around 1 s to *mid-air pen* is similar to the difference found in our previous study [30].

Figure 5 (bottom) shows that even after the menu is opened, *surface* has the slowest selection time compared to the baseline (*two-handed touch*, M: 1.37 s, CI [1.2, 1.6]). *One-handed touch* seems

[3]https://hci.rwth-aachen.de/armenus (accessed June 12, 2020)

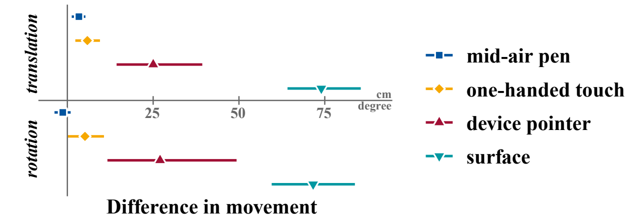

Figure 6: Difference in *translation* (in cm) and *rotation* (in degrees) compared to the baseline. *Surface* shows the most additional movement while *mid-air pen* and *one-handed touch* seem to be close to the baseline. Whiskers denote the 95% CI.

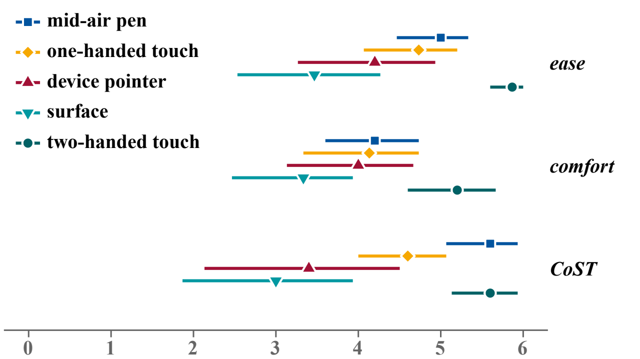

Figure 7: Subjective ratings for *ease* (top), *comfort* (middle), and combination of selection techniques (*CoST*, bottom). *Two-handed touch* has the highest ratings while *surface* seems to be less easy and comfortable to use. Whiskers denote the 95% CI.

to be close to the performance of the baseline with *mid-air pen* and *device pointer* following.

Regarding the movement of the device after opening the menu, *translation* (Fig. 6, top) and *rotation* (Fig. 6, bottom) show that, as expected, *surface* shows the most movement of the techniques. *Device pointer* measurements show that, compared to the baseline, participants used both more rotation and translation movement. This indicates that not one type of movement was singularly used. Both *mid-air pen* and *one-handed touch* seem to be similar in the required movement of the device compared with the baseline with a trend to more translation. On average, participants translated the device by 12.6 cm (CI [10.6, 14.6]) and rotated the device by 19.9 degrees (CI [17.8, 22.6]) in the baseline condition.

4.5.2 Ease, Comfort, Combination of Selection Techniques, and Ranking. Looking at the subjective ratings for ease of use of the techniques (Fig. 7, top), we see that the *two-handed touch* baseline was rated as the easiest technique to use. The other techniques show ratings similar to each other but the tendency goes towards the order *mid-air pen*, *one-handed touch*, *device pointer*, and *surface*. Ratings regarding the comfort are even closer together than the ratings for ease of use (Fig. 7, middle). Especially *mid-air pen*,

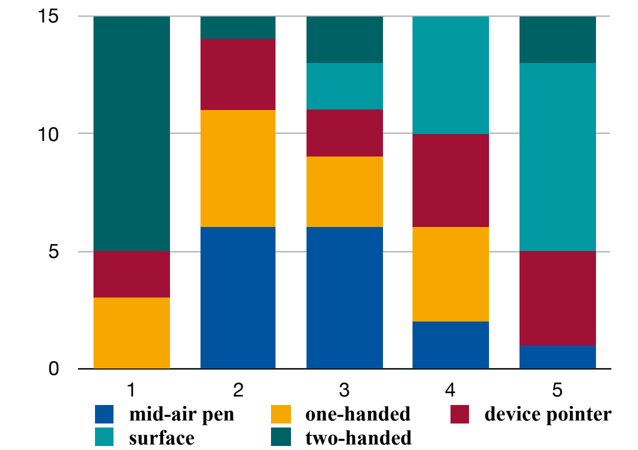

Figure 8: Subjective ranking of the techniques. *Two-handed touch* seems to be most preferred while *surface* is often placed in the last position. *One-handed touch* and *mid-air pen* are placed high and *device pointer* is placed at all positions in the ranking.

one-handed touch, and *device pointer* move closer together and also *two-handed touch* was rated more similar to them while still being rated more comfortable. The results of asking participants to rate the combination of selection techniques (*CoST*) to open the menu and select the item shows that the techniques that use the same method for both (*two-handed touch*, *mid-air pen*) achieve the highest ratings (Fig. 7, bottom) followed by *one-handed touch*. *Device pointer* and *surface* received lower ratings.

The ranking of techniques shows that participants seem to prefer the baseline condition (*two-handed touch*) while placing *surface* often in the last place (Fig. 8). Both *one-handed touch* and *mid-air pen* are placed more towards the top of the ranking while *device pointer* was placed at every place of the ranking.

4.5.3 Qualitative Remarks. While *surface* seemed to be less liked by participants, several mentioned that they liked the physical guidance of the surface but only two rested their hand on the surface while approaching the target item. However, participants commented that they did not like the switch in context between mid-air target and menu on the surface. For *device pointer*, seven participants explained confusion about which device/cursor to use for which step even though only one cursor was visible at any time. On the other hand, other participants assumed that the performance of this technique could improve a lot when using it for a longer time, with three participants explicitly stating that the learning process was very fast. The biggest commented issue for *one-handed touch* was that participants had trouble holding the phone in the grip and simultaneously using the thumb to interact. Others, however, mentioned that they liked that the menu is directly under the thumb and that the shape would help to select the menu items. For *one-handed touch*, *two-handed touch*, and *mid-air pen*, participants said that they found the techniques easy to use and especially *two-handed touch* would be what they are used to. However, two participants mentioned for *two-handed touch* that it would be annoying to switch between mid-air interaction with the pen and touchscreen interaction frequently. Similar to our previous studies [30, 32], participants noted that they felt fatigue in their arms since they had to hold both smartphone and pen in mid-air for most parts of the interaction. This shows that this interaction is not suitable for prolonged use but more for short periods of time.

5 DISCUSSION

Our results indicate that the baseline condition of *two-handed touch* is the fastest and most preferred technique to use to interact with menus in 3D environments on a handheld device. This is interesting in comparison to our previous selection study [30] in which the pen-ray interaction was more similar to the two-handed interaction. On the other hand, the one-handed interaction was not rated high in the selection study but achieved good results and ratings in our study. This means that fixing the menu to the side of the screen improves this technique. *Mid-air pen* and *one-handed touch* are not far behind *two-handed touch* as they showed not only similar success rates but did not differ much in terms of the device movement and also achieved sufficiently close selection times. Only for the time to open a menu, *two-handed touch* seems to be about 1 s faster which is in line with the findings from [30]. In a use case where a lot of the interaction is happening mid-air, such as sketching or manipulating objects, frequent switches between mid-air and touchscreen interaction will limit this time benefit of *two-handed touch*. In such cases, staying within the context in mid-air could prove to be advantageous. *One-handed touch* and *mid-air pen* performed largely comparable. The differences in the time to open the menu between the techniques that use the pen-ray selection to trigger the menu can be explained by the different behavior after an item has been selected and the new target cube is highlighted: For both *one-handed touch* and *device pointer*, the focus of the user is not on the mid-air pen, requiring the user to find the pen tip again. For *surface*, participants had to lift the phone from the surface to see the mid-air cubes again. While users preferred using the same method for opening the menu and selecting an item in *mid-air pen*, *one-handed touch* seems to be a bit faster for selecting the menu item and it can be used independently of the pen position in mid-air. This could be advantageous for more complex menu input so that users can rest their hands and are not required to keep them lifted during the whole interaction. This point could also be a benefit for *surface* which did not perform as well as other techniques in our metrics. We expected that this technique requires more movement and time for the interaction due to the world fixed style of the menu and also since it is the only menu technique that requires a depth estimation by the user to move to the intended location. On the other hand, the physical surface provides haptic feedback during the interaction and we wanted to see whether this could compete with the movement and time increases. While participants mentioned that they liked the physical guidance of the surface, they gave this condition lower ratings and a lower ranking. This menu type might be helpful to use in scenarios where opening and closing the menu does not happen frequently and when the menu interaction is followed by an interaction for which the physical surface provides an advantage. An example could be the situation where a hand drawing should be projected on a plane or object in the scene. Selecting the object mid-air could then lead to the menu specifying the drawing properties using the surface menu before using the physical surface to draw. *Device pointer* achieved mixed results indicated also in the spread across all placements in the ranking. While success rate and interaction times are comparable or close behind the other techniques, using this techniques requires more movement of the device and participants mentioned that it is more difficult to understand this technique. On the other hand participants also mentioned that the interaction would be fast to learn. Since it does not require to move the pen or the focus of the user, it could be an interesting technique for routine and quick selections that can then be selected blindly by ‘flicking' the device. An example could be to choose that the selected object should be moved. After selecting this option with *device pointer*, the object snaps to the pen tip so that it can be placed.

For menus, there is no "one size fits all", and designers will have to weigh the different options for their specific scenario. Our results allow for them to have a starting point, e.g., to decide whether a ∼1s increase in item selection time is acceptable if the device pointer otherwise fits their interaction scenario. The measurements do not directly disqualify any menu technique and for each there could be a scenario where it is the most sensible choice. Following, we present general suggestions as well as potential use cases, based also on the qualitative feedback from our participants, in which choosing a menu technique other than the "fastest" or "most accurate" could make sense:

- If the interaction happens on the touchscreen and only occasionally mid-air, use the standard *two-handed touch*.
- If most of the interaction happens mid-air, consider using *one-handed touch* and *mid-air pen*. The preference for either depends on the use case.
- For quick and routine actions, consider *device pointer*. Users will be able to keep the mid-air pen and their focus at their current position.
- If the interaction after using the menu benefits from a physical surface, using *surface* provides the benefits from the haptic surface, which could outweigh its drawbacks such as longer interaction times.

Our study provides first insights into the effect of different interaction methods on menu performance in Handheld AR systems. To encourage further research and replication, we provide both our software and data with this submission[4].

6 SUMMARY & FUTURE WORK

We present the results of a study comparing menu techniques for Handheld AR applications using a mid-air pointing device. The menu techniques were sampled to account for the different interaction methods such a system offers: mid-air, touchscreen, physical environment, and movement of the device itself. We found that the standard technique of *two-handed touch* seems to achieve the best results overall. However, if most of the interaction happens in mid-air, switching to the touchscreen can become annoying. In such cases, a menu in mid-air (*mid-air pen*) or a menu on the side of the touchscreen to be operated by the hand holding the smartphone (*one-handed touch*) seems to be the preferable options. Our results provide interaction designers with an estimation of the

[4]https://hci.rwth-aachen.de/armenus (accessed June 12, 2020)

differences between the menu techniques so that they can pick the most suitable for their application.

We have used a simple scenario for our study to evaluate the general impact of the different interaction methods on menu performance and reduce the impact of extraneous factors as much as possible. Future studies could increase the external validity by studying the menu techniques in scenarios closer to the real world. This could mean different surroundings, such as cars and kitchens, as well as different tasks, such as menu interactions embedded in a modeling or drawing task. While we focused on menus that use the main interaction capabilities of a Handheld AR system with a mid-air pen, there are numerous ways how such a system could be extended. These extensions, for example back-of-device or voice interaction, could also be used for menu interactions and we encourage further exploration in this direction. Other studies in this area could adjust different aspects specified by the taxonomy from Dachselt and Hübner [8], and study their impact. An interesting direction, for example, is to adjust the hierarchical nature to require multiple menu levels and the navigation between them—a possible visualization for additional menu levels is discussed by Gebhardt et al. [14]. Other promising research questions include matching different menu techniques to specific tasks such as browsing through 3D objects, and then combining these different menus in a coherent and usable way.

REFERENCES

[1] Rahul Arora, Rubaiat Habib Kazi, Fraser Anderson, Tovi Grossman, Karan Singh, and George Fitzmaurice. 2017. Experimental Evaluation of Sketching on Surfaces in VR. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. ACM, New York, NY, USA, 5643–5654. https://doi.org/10.1145/3025453.3025474

[2] Huidong Bai, Gun A. Lee, Mukundan Ramakrishnan, and Mark Billinghurst. 2014. 3D Gesture Interaction for Handheld Augmented Reality. In *SIGGRAPH Asia 2014 Mobile Graphics and Interactive Applications* (Shenzhen, China) *(SA '14)*. ACM, New York, NY, USA, Article 7, 6 pages. https://doi.org/10.1145/2669062.2669073

[3] Mark Billinghurst, Tham Piumsomboon, and Huidong Bai. 2014. Hands in Space: Gesture Interaction with Augmented-Reality Interfaces. *IEEE Computer Graphics and Applications* 34, 1 (Jan 2014), 77–80. https://doi.org/10.1109/MCG.2014.8

[4] Doug A. Bowman and Chadwick A. Wingrave. 2001. Design and Evaluation of Menu Systems for Immersive Virtual Environments. In *Proceedings IEEE Virtual Reality 2001*. IEEE, New York, NY, USA, 149–156. https://doi.org/10.1109/VR.2001.913781

[5] Jack Callahan, Don Hopkins, Mark Weiser, and Ben Shneiderman. 1988. An Empirical Comparison of Pie vs. Linear Menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Washington, D.C., USA) *(CHI '88)*. ACM, New York, NY, USA, 95–100. https://doi.org/10.1145/57167.57182

[6] James Carpenter and John Bithell. 2000. Bootstrap confidence intervals: when, which, what? A practical guide for medical statisticians. *Statistics in Medicine* 19, 9 (2000), 1141–1164. https://doi.org/10.1002/(SICI)1097-0258(20000515)19:9<1141::AID-SIM479>3.0.CO;2-F

[7] Geoff Cumming. 2014. The New Statistics: Why and How. *Psychological Science* 25, 1 (2014), 7–29. https://doi.org/10.1177/0956797613504966

[8] Raimund Dachselt and Anett Hübner. 2007. Virtual Environments: Three-dimensional Menus: A Survey and Taxonomy. *Comput. Graph.* 31, 1 (Jan. 2007), 53–65. https://doi.org/10.1016/j.cag.2006.09.006

[9] Nguyen-Thong Dang and Daniel Mestre. 2011. Effects of Menu Orientation on Pointing Behavior in Virtual Environments. In *Proceedings of the 2011 International Conference on Virtual and Mixed Reality: New Trends - Volume Part I* (Orlando, FL, USA). Springer-Verlag, Berlin, Germany, 144–153. http://dl.acm.org/citation.cfm?id=2028650.2028669

[10] Kaushik Das and Christoph W. Borst. 2010. An evaluation of menu properties and pointing techniques in a projection-based VR environment. In *2010 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, New York, NY, USA, 47–50. https://doi.org/10.1109/3DUI.2010.5444721

[11] Pierre Dragicevic. 2016. *Fair Statistical Communication in HCI*. Springer International Publishing, Cham, Switzerland, 291–330. https://doi.org/10.1007/978-3-319-26633-6_13

[12] Bradley Efron. 1987. Better Bootstrap Confidence Intervals. *J. Amer. Statist. Assoc.* 82, 397 (1987), 171–185. https://doi.org/10.1080/01621459.1987.10478410

[13] Steven Feiner, Blair MacIntyre, Marcus Haupt, and Eliot Solomon. 1993. Windows on the World: 2D Windows for 3D Augmented Reality. In *Proceedings of the 6th Annual ACM Symposium on User Interface Software and Technology* (Atlanta, Georgia, USA) *(UIST '93)*. ACM, New York, NY, USA, 145–155. https://doi.org/10.1145/168642.168657

[14] Sascha Gebhardt, Sebastian Pick, Franziska Leithold, Bernd Hentschel, and Torsten Kuhlen. 2013. Extended Pie Menus for Immersive Virtual Environments. *IEEE Transactions on Visualization and Computer Graphics* 19, 4 (April 2013), 644–651. https://doi.org/10.1109/TVCG.2013.31

[15] Richard H. Jacoby and Stephen Ellis. 1992. Using virtual menus in a virtual environment. *Proceedings of SPIE - The International Society for Optical Engineering* 1668 (02 1992), 39–48. https://doi.org/10.1117/12.59654

[16] Ke Huo, Vinayak, and Karthik Ramani. 2017. Window-Shaping: 3D Design Ideation by Creating on, Borrowing from, and Looking at the Physical World. In *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction* (Yokohama, Japan) *(TEI '17)*. ACM, New York, NY, USA, 37–45. https://doi.org/10.1145/3024969.3024995

[17] Transparent Statistics in Human-Computer Interaction Working Group. 2019. Transparent Statistics Guidelines. https://doi.org/10.5281/zenodo.1186169 (Available at https://transparentstats.github.io/guidelines).

[18] John I. Kiger. 1984. The Depth/Breadth Trade-off in the Design of Menu-driven User Interfaces. *Int. J. Man-Mach. Stud.* 20, 2 (March 1984), 201–213. https://doi.org/10.1016/S0020-7373(84)80018-8

[19] Namgyu Kim, Gerard Jounghyun Kim, Chan-Mo Park, Inseok Lee, and Sung H. Lim. 2000. Multimodal Menu Presentation and Selection in Immersive Virtual Environments. In *Proceedings of the IEEE Virtual Reality 2000 Conference (VR '00)*. IEEE Computer Society, Washington, DC, USA, 281–. http://dl.acm.org/citation.cfm?id=832288.835758

[20] Rick Komerska and Colin Ware. 2004. A Study of Haptic Linear and Pie Menus in a 3D Fish Tank VR Environment. In *Proceedings of the 12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS '04)*. IEEE, New York, NY, USA, 224–231. https://doi.org/10.1109/HAPTIC.2004.1287200

[21] Ernst Kruijff, J. Edward Swan, and Steven Feiner. 2010. Perceptual issues in augmented reality revisited. In *2010 IEEE International Symposium on Mixed and Augmented Reality*. IEEE, New York, NY, USA, 3–12. https://doi.org/10.1109/ISMAR.2010.5643530

[22] T. K. Landauer and D. W. Nachbar. 1985. Selection from Alphabetic and Numeric Menu Trees Using a Touch Screen: Breadth, Depth, and Width. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Francisco, California, USA) *(CHI '85)*. ACM, New York, NY, USA, 73–78. https://doi.org/10.1145/317456.317470

[23] Hyeongmook Lee, Dongchul Kim, and Woontack Woo. 2011. Graphical Menus Using a Mobile Phone for Wearable AR Systems. In *Proceedings of the 2011 International Symposium on Ubiquitous Virtual Reality (ISUVR '11)*. IEEE Computer Society, Washington, DC, USA, 55–58. https://doi.org/10.1109/ISUVR.2011.23

[24] Dwight P. Miller. 1981. The Depth/Breadth Tradeoff in Hierarchical Computer Menus. *Proceedings of the Human Factors Society Annual Meeting* 25, 1 (1981), 296–300. https://doi.org/10.1177/107118138102500179

[25] Annette Mossel, Benjamin Venditti, and Hannes Kaufmann. 2013. 3DTouch and HOMER-S: Intuitive Manipulation Techniques for One-handed Handheld Augmented Reality. In *Proceedings of the Virtual Reality International Conference: Laval Virtual* (Laval, France) *(VRIC '13)*. ACM, New York, NY, USA, Article 12, 10 pages. https://doi.org/10.1145/2466816.2466829

[26] Annette Mossel, Benjamin Venditti, and Hannes Kaufmann. 2013. DrillSample: Precise Selection in Dense Handheld Augmented Reality Environments. In *Proceedings of the Virtual Reality International Conference: Laval Virtual* (Laval, France) *(VRIC '13)*. ACM, New York, NY, USA, Article 10, 10 pages. https://doi.org/10.1145/2466816.2466827

[27] Jarkko Polvi, Takafumi Taketomi, Goshiro Yamamoto, Arindam Dey, Christian Sandor, and Hirokazu Kato. 2016. SlidAR: A 3D positioning method for SLAM-based handheld augmented reality. *Computers & Graphics* 55 (2016), 33 – 43. https://doi.org/10.1016/j.cag.2015.10.013

[28] Ron van Teylingen, William Ribarsky, and Charles van der Mast. 1997. Virtual Data Visualizer. *IEEE Transactions on Visualization and Computer Graphics* 3, 1 (Jan 1997), 65–74. https://doi.org/10.1109/2945.582350

[29] Thomas Vincent, Laurence Nigay, and Takeshi Kurata. 2013. Precise Pointing Techniques for Handheld Augmented Reality. In *Human-Computer Interaction – INTERACT 2013*. Springer Berlin Heidelberg, Berlin, Heidelberg, 122–139.

[30] Philipp Wacker, Oliver Nowak, Simon Voelker, and Jan Borchers. 2019. ARPen: Mid-Air Object Manipulation Techniques for a Bimanual AR System with Pen & Smartphone. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. ACM, New York, NY, USA, Article 619, 12 pages. https://doi.org/10.1145/3290605.3300849

[31] Philipp Wacker, Adrian Wagner, Simon Voelker, and Jan Borchers. 2018. Physical Guides: An Analysis of 3D Sketching Performance on Physical Objects

in Augmented Reality. In *Proceedings of the Symposium on Spatial User Interaction* (Berlin, Germany) *(SUI '18)*. ACM, New York, NY, USA, 25–35. https://doi.org/10.1145/3267782.3267788

[32] Philipp Wacker, Adrian Wagner, Simon Voelker, and Jan Borchers. 2020. Heatmaps, Shadows, Bubbles, Rays: Comparing Mid-Air Pen Position Visualizations in Handheld AR. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3313831.3376848

[33] Po-Chen Wu, Robert Wang, Kenrick Kin, Christopher Twigg, Shangchen Han, Ming-Hsuan Yang, and Shao-Yi Chien. 2017. DodecaPen: Accurate 6DoF Tracking of a Passive Stylus. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Quebec City, QC, Canada) *(UIST '17)*. ACM, New York, NY, USA, 365–374. https://doi.org/10.1145/3126594.3126664

[34] Min Xin, Ehud Sharlin, and Mario Costa Sousa. 2008. Napkin Sketch: Handheld Mixed Reality 3D Sketching. In *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology* (Bordeaux, France) *(VRST '08)*. Association for Computing Machinery, New York, NY, USA, 223–226. https://doi.org/10.1145/1450579.1450627