Interaction Techniques for 3D-positioning Objects in Mobile Augmented Reality

Carl-Philipp Hellmuth
Bayreuth University
Germany
Carl-Philipp.Hellmuth@uni-bayreuth.de

Miroslav Bachinski
Bayreuth University
Germany
Miroslav.Bachinski@uni-bayreuth.de

Jörg Müller
Bayreuth University
Germany
Joerg.Mueller@uni-bayreuth.de

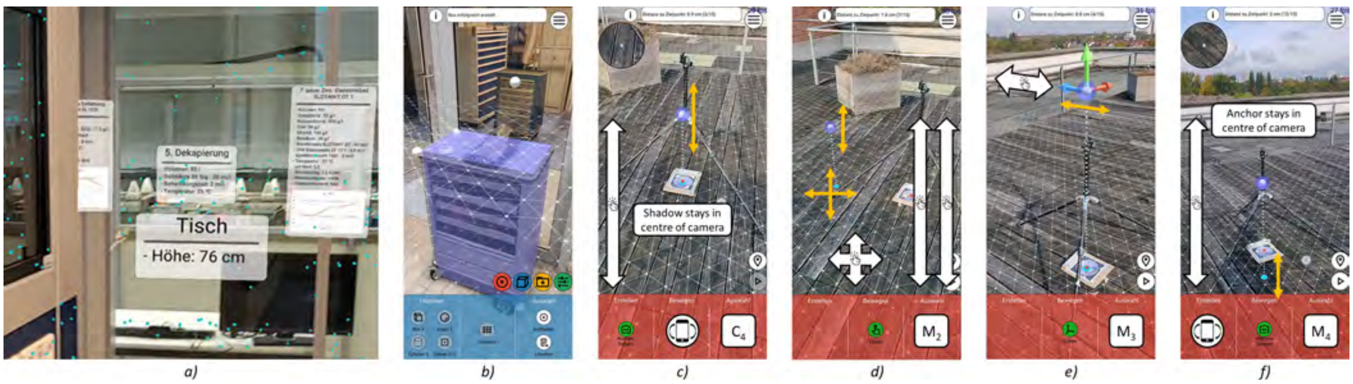

Figure 1: a) a partially occluded data sheet behind a wall on the left; b) a virtual box approximating a real object; c) to f) different interaction methods with annotations: The big white arrows are the touch movements (only the direction matters ,except for "e)") and the small yellow arrows symbolise the movement of the anchor or the shadow of the anchor.

ABSTRACT

This paper explores interaction techniques for positioning objects in 3D-space during instantiation and movement interactions in mobile augmented reality. We designed the methods for 3D-objects positioning (no rotation or scaling) based on camera position and orientation, touch interaction, and combinations of these modalities. We consider four interaction techniques for creation: three new techniques and an existing one, and four techniques for moving: two new techniques and another two from previous work. We implemented all interaction methods within a smartphone application and used it as a basis for the experimental evaluation. We evaluated the interaction methods in a comparative user study (N=12): The touch-based methods outperform the camera-based techniques in perceived workload and accuracy. Both are comparable regarding the task completion time. The multimodal methods performed worse than the methods based on individual modalities both in terms of performance and workload. We discuss the implications of these findings to the HCI research and provide corresponding design recommendations. For example, we recommend avoiding the combination of camera and touch-based methods for a simultaneous interaction, as they interfere with each other and introduce jitter and inaccuracies in the user input.

ICMI '21, October 18–22, 2021, Montréal, QC, Canada

ACM ISBN 978-1-4503-8481-0/21/10…$15.00
https://doi.org/10.1145/3462244.3479945

CCS CONCEPTS

• **Human-centered computing** → **Mixed / augmented reality**; *User studies*; **Touch screens**; **Smartphones**; *Information visualization.*

KEYWORDS

AR Interaction Techniques, 3D Manipulation, mobile Augmented Reality, Occlusion, AR in industry

ACM Reference Format:

Carl-Philipp Hellmuth, Miroslav Bachinski, and Jörg Müller. 2021. Interaction Techniques for 3D-positioning Objects in Mobile Augmented Reality. In *Proceedings of the 2021 International Conference on Multimodal Interaction (ICMI '21), October 18–22, 2021, Montréal, QC, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3462244.3479945

1 INTRODUCTION

One of the significant strengths of augmented reality (AR) is the possibility to show digital information to the users based on their spatial position, and viewpoint [2]. This quality has a high potential in various contexts, for example, in an industrial environment where different data sources generate immense amounts of data that must be presented to the users [27]. However, the users can only benefit from the data if it is structured and visualised not to overwhelm them [28]. AR allows showing the data according to its spatial relevance, for example, at the device or component that created it. If the data is arranged spatially, the users only observe the most relevant data for a given spatial context.

Another step to avoid visual clutter is to correctly occlude the digital content by the 3D geometry of the physical environment. This geometry can be either identified at runtime via a depth sensor [7] or a camera [5], or stored on the device as a static environment model [9]. The first approach is cumbersome to use, has a limited range not suitable for large spaces, such as a factory, and poorly represents complex shapes, for instance production machines [12]. Thus the second option is more practical, namely enabling the users to approximate the environment with geometric primitives. Among the ways to create geometric primitives, we consider anchor-based approach, namely constructing complex objects from a selection of 3D anchor points positioned in the environment, as the most flexible and better fitting our use-case, although requiring more work from users. This approach is also more consistent with respect to the interactions, as users can apply the same methods to position both anchors and data visualisations in the 3D space. In this way, the complete 3D scene is defined by the placement of the anchors.

To make spatial arrangement of the data visualisations and the anchor-based geometry generation usable, we require precise and efficient interaction techniques to instantiate or move an anchor or digital object to a specific 3D position in the environment. For positioning the 3D point anchors we require translational and can ignore rotational degrees of freedom. While there are typical interaction techniques for the placement and manipulation of objects in the 3D environment for desktop UI, the design of interaction techniques suitable for AR on a handheld device with a small touch screen remains unclear and challenging [10]. We use 2 modalities available on smartphones: touchscreen and movement of the device, as well as their combination to design 3 new interaction techniques to create anchors and another 2 new interaction techniques to move them in the 3D environment. We implement these and 3 other interaction techniques from the literature in our application and evaluate user performance, perceived workload, and preference when interacting with them. We use NASA-TLX [14], additional interaction-specific questions, and multiple quantitative metrics based on the data collected on the device. We compare the different techniques for placement and movement against each other and derive recommendations for using these methods in similar use cases.

To sum up, this paper makes the following contributions:

- we design 5 new interaction techniques for the placement and movement of 3D objects in a handheld AR environment,
- we evaluate and compare these techniques and 3 techniques from the literature according to their performance and perceived workload in a user study,
- we provide recommendations for the design of such interactions for similar use-cases in mobile AR.

2 RELATED WORK

According to the "Review of 3D object manipulation techniques in Handheld mobile AR-Interfaces" from Goh Et al. [10] it is possible to sort mobile AR-Interfaces into three categories: Touch-based, Device-based, and Mid-Air Gesture-based. In this paper, we abstain from treating mid-air gesture-based interactions, as they entail a very own set of challenges and benefits and therefore cannot be compared with the two other techniques this study deals with. For example, for performing a mid-air gesture-based interaction the user must hold the device with one hand while keeping the other hand within the camera view; this drastically shrinks the space where the user can interact [16].

2.1 Touch-based

For smartphone-based AR applications, it seems obvious to use the touch screen as the input method. A touch screen allows many interaction techniques for controlling digital content, and it is already part of everyday smartphone usage [10]. In the context of AR, it makes even more sense, as the display is the only place where the digital objects are visible for the users. In most cases, the manipulation of 3D objects in AR is not that different from manipulating the same objects in a completely digital 3D environment. This topic is well-researched offering many solutions [13, 19, 20]. But also for AR interactions, a range of techniques exist. There are methods just focused on the placement of objects, like the methods demonstrated in the DepthLab demo [5], or complete systems, ranging from 3DTouch [22] where users have to select whether to translate, rotate or scale the object, to systems like DS3 [21] where the gesture type selects the manipulation action. Both of the last approaches and all approaches between try to solve the problem of mapping 6 degrees of freedom (DOF) onto a 2D Surface. The approach of selecting the action based on the finger movement allows the users to quickly switch between actions to finely adjust an object or manipulate many objects in quick succession [19]. The possibility to select the manipulation type allows the gestures to be simpler. Furthermore, the gestures can be easier to understand. They do not have to be distinct from each other since they do not need to be differentiated by the system, making it potentially easier to pick up. Methods like the focal point placement in the DepthLab demo [5] allow the placement of an anchor with one touch.

2.2 Device-based

In the case of device-based interactions, the device is not only used as a window to see virtual objects but also as a tool to manipulate them [10]. A study by Tanikawa et al. [26] is an excellent example of this technique: the user removed blocks from a virtual Jenga tower in AR by attaching the block to a virtual stick extending from the device. After attaching the block, it follows the translations and rotations of the device and keeps the same position and rotation relative to the device (0 DOF).

In contrast, with the system HOMER-S [22], the user selects a manipulation action to translate, rotate or scale the object by moving the device. In the evaluation comparing HOMER-S with the gizmo based 3DTouch controls published in the same paper, HOMER-S achieved better performance and ease of use rating for positioning or rotating tasks, but for scaling the 3DTouch was the clear winner [22]. Those results match with the explorative study of Dong et al. [4], where the touch and device-based interactions methods excelled at different tasks. Overall, the device-based methods were rated more engaging, especially when the device movement matched the motion for the real-world task (e.g., stretching and aiming a slingshot). A problem that may often appear while using a device-based interaction for rotating or scaling is that the users might have to move the device into a position where they do not see the object anymore. To avoid this, the action has to be easily interruptible, allowing the user to reposition the device and continue manipulating the object [24, 26].

2.3 Multimodal

Multimodal interactions or interfaces use elements from different interaction modalities combining them into a single interaction [8]. For example, the combination of hand gesture inputs (via hand tracking, gloves or other hardware) with voice commands was studied in multiple scenarios and achieved good results [17, 18, 23]. A different example is the use of touch and voice inputs to define (touch) and name (voice) objects in a screenshot from a camera feed in "In situ CAD Capture" [25].

In this paper we describe multimodal interactions that combine device and touch-based inputs as this allows us to build on our work of the non-multimodal interactions. We define that using the touch screen for selecting an object or the manipulation action while using a different input modality for the manipulation does not qualify as a multimodal interaction. Most of the previous interaction techniques are designed to manipulate complex 3D objects. They are excessively complicated for the 3D-point based system, so we chose 3 interaction methods [5, 22, 26] that fit our use case the best and designed 5 new interaction methods, which focus on the creation and movement of 3D-points. In the user study, we compared the qualities of all selected interaction techniques in the context of the primary use-case of our system.

3 GENERAL DESIGN CONSIDERATIONS

The interaction techniques were implemented within an AR-based interactive data visualisation application. The application was designed for an industrial environment in close collaboration and with feedback from our industry partners. The system's goals and the requirements were established based on interviews with the potential users. They had a significant impact on the design choices and the considered interaction techniques.

The application was designed in an iterative design process with regular feedback from the potential users. It has converged on the following key requirements and design decisions:

- **The form-factor** was considered to be mobile AR or a head-mounted AR display (HMD). However, the HMD as a platform was discarded based on the feedback of industry partners, with the main concerns being the high price and the fragility of the devices compared to smartphones.
- **The tracking technology** was chosen to be the state-of-the-art marker-less tracking of ARCore. Marker-based tracking was not an option, as placing all necessary markers in a factory would be very time consuming [6]. Radio or magnetic signal-based tracking methods are not feasible at the factories and plants, as the movement of large metal components and materials significantly distorts the signal of Bluetooth or Wi-Fi beacons on a given point [1].
- **The software platform** was selected to be Unity as a widely used universal cross-platform rendering engine.
- **Modes of interaction** for the application were considered to be visualiser and editor. In the visualiser mode, the user view shows the data with the correct occlusion. The editor mode gives users the possibility to place anchors, connect data entries to those anchors, and define simple volumes with these anchors to approximate real-world geometry.
- **Interaction techniques** need to provide an intuitive, flexible, and efficient way to create and precisely position the anchors in the whole 3D industrial AR environment when using the application in editor mode.
- **The information hierarchy** needs to be available to users to select the required data sources and link them to the given anchors at the correct position in the 3D space (Figure 1 a)).
- **Improve depth cues** by dropping shadow below a moving anchor based on Diaz et al. [3], which allows users to quickly judge the distance to a hovering anchor based on its virtual shadow on the ground (Figure 1 f)).
- **Geometric primitives** such as a sphere, cuboid, and cylinder need to be intuitively constructed or modified based on a small set of selected anchors. They need to occlude the data visualisations in the visualiser mode.

4 INTERACTION TECHNIQUES

The interaction methods for anchor creation and manipulation were designed and implemented in the high-level prototype of the application. We chose 3 existing interaction techniques because they fit our use case and achieved good results in the previous works [5, 22, 26]. But most other interactions in the literature focus on moving, rotating and scaling objects with one set of methods. This leads to unnecessary complexity in our use case, so we developed 5 new methods. We implemented 4 interaction methods to create an anchor and 4 methods to move it. This section provides the details of each interaction method, including the action sequence in parenthesis. Each method has to be activated by pressing a button in the menu section at the bottom of the screen (counted as step 0). The movement methods have to be stopped with the same button.

4.1 Creation methods

4.1.1 Create an anchor at the camera (C_1, device-based, novel): by (1) pushing the corresponding button in the menu section at the bottom of the screen, users immediately create an anchor at the device's position. To improve the usability of this device-based method, an offset is applied to place the anchor 2.5 cm in front of the camera. This gives users instant visual feedback because the anchor fills the whole screen and is not blocked by the near clipping plane of the virtual camera. This method should allow users to rapidly place anchors by pressing just one button without judging the distance to some point, as is the case with some other methods. But it only allows the placement of anchors at positions that users can physically reach (e.g. it can not be used for tall objects or tight spaces) and where users do not endanger themselves (e.g. getting too close to dangerous machinery).

4.1.2 Create an anchor on the surface (C_2, touch-based, existing): users can place an anchor on a surface detected by ARCore or a surface of a geometric primitive by (1) touching the position on the screen. This method is used in the examples of ARCore [11] and some commercial apps (e.g. Civilisations AR). It works similarly to the object placement method of the DepthLab Demo but does not rely on a 3D-cursor [5]. While the screen is pressed (2), a magnified

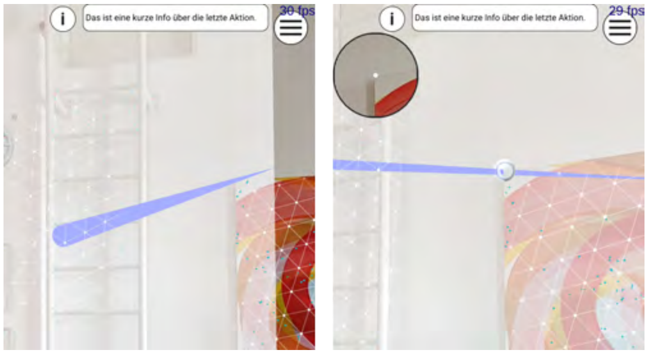

Figure 2: The left image shows the first ray and the half-transparent plane. In the right image, the user aims again for the corner of the painting using the magnification view.

view of the camera image around the finger is shown in the top left corner to allow for more precise placement. Finally, (3) the anchor is placed by releasing the touch. With this interaction, users should be able to intuitively place anchors on the floor or the wall, if ARCore correctly detects the surface. Moreover, this method allows users to place anchors with high precision at small to medium distances (depending on the user's fine motor skills) but only supports 2 DOF, as it is bound to a surface.

4.1.3 Create an anchor at the intersection (C_3, multimodal, novel): users create a ray by (1) touching a point in the screen environment (the magnification view is enabled). The ray is visualised with a blue line and permanently added to the scene going from the position of the device at the time of creation through the point in the environment (Figure 2(left)). After (2) a change of viewpoint (e.g. two steps to the side), users can (3) touch the same point in the environment on the screen again to cast a second ray. At the intersection point of both rays, an anchor is created (Figure 2(right)), and the rays are removed. To allow users to easily hit the first ray with the second one, a semi-transparent vertical plane is added to the first ray. This method adds the third dimension to C_2 at the cost of more interaction steps and requiring two precise inputs instead of one. It also isn't bound by the surface detection of ARCore, so any point, even at greater distances ($> 10m$), can be targeted.

4.1.4 Create an anchor in front of the camera (C_4, multimodal, novel): This multimodal method creates an anchor as soon as the touch (1) starts; the anchor (2 a) follows the camera until the touch is (3) released. The drop shadow of the anchor is always placed at the intersection of a ray cast through the centre of the camera with the ground plane detected by ARCore. By (2 b) moving the finger up and down on the screen, users can adjust the anchor's height above the drop shadow (Figure 1 c). This method expands the concept used by some commercial AR apps (e.g. Minecraft AR) where users see a preview of the object they want to place at the intersection between the closest detected surface and a ray through the camera centre. Then, when users touch the object, it is anchored to the current position. We didn't use this basic method as it is even more restricted than C_2. But by combining the touch input with the device-based input, we added the third dimension and created a method that distinctively differs from the basic method.

4.2 Movement Methods

4.2.1 Move an anchor with the camera (M_1, device-based, existing): This device-based method (1) fixes the anchor to the device (0 DOF) until the process is (2) disabled by users [26]. In this way, users can carry the anchor around or adjust the position while using the drop shadow as a guide to judge the depth. This method is proposed and evaluated in multiple papers [22, 26] because it is similar to the way a user interacts with objects in reality: grabbing the object with one hand (pushing a button on the screen) – > moving the hand to move the object (moving the device) – > releasing the object (pushing the button again). The cited studies focus on the interaction with objects at a short distance, where this method performs well [22]. The technique retains its input precision for manipulations at a greater distance ($1cm$ device movement is always $1cm$ anchor movement). However, it becomes difficult to see and judge the position of the anchor at greater distances.

4.2.2 Move an anchor with the finger (M_2, touch-based, novel): after activating this method, users can (1 a) drag one finger on the screen to move the anchor parallel to the ground plane. By (1 b) dragging two fingers up or down on the screen, users can adjust the height (Figure 1 d). By splitting the manipulation of the 3 DOF into two separate input gestures, we try to reduce the complexity of the method. This allows users to focus on placing the drop shadow at the target position and then concentrate on adjusting the anchor's height.

4.2.3 Move an anchor with the 3D-movement widget (M_3, touch-based, existing): this method draws a 3-arrow widget aligned with the base coordinate system's axis at the anchor position when activated. To move the anchor on the axis, users have to (1) touch and (2) drag the corresponding arrowhead of the widget (Figure 1 e). This method is an implementation of the standard manipulation technique for 3D objects in productive desktop software. A similar approach is used in the translation interaction of the 3DTouch technique where users swipe along the axes of a coordinate system centred on the object [22]. We decided to stay closer to the classical implementation because in the evaluation of 3DTouch, their implementation performed similar to their implementation of M_1, and we wanted a better contrast to M_2.

4.2.4 Move an anchor with distance to camera (M_4, multimodal, novel): This multimodal method works similar to the C_2 method. While the method is activated, the anchor snaps to the centre of the screen and stays there for the movement duration while always keeping the same distance to the device on the ground plane. The (1 a) device-based inputs move the anchor along the surface of an imaginary cylinder. The up or down (1 b) movement of one finger on the screen increases or decreases the cylinder's radius, in other words the shadow of the anchor is moved closer or father away from the camera (Figure 1 f). This method splits the 3 DOF between the device and touch-based inputs. This allows users to first adjust the distance of the anchor to the device (touch input) and then place the anchor at the correct height and horizontal position.

5 USER STUDY

We conducted a user study to subjectively evaluate the system and compare the individual interaction techniques concerning perceived workload, task performance and accuracy.

5.1 Experimental design

The experiment consisted of three parts: evaluating the anchor creation methods, assessing the anchor movement methods, and subjective evaluation of the 3D geometry manipulation.

In the first part of the experiment, the 4 experimental conditions were defined according to the creation methods. The order of the conditions was counterbalanced by a Latin square. The task was to create new anchors at specified points in the 3D space. The target points were marked by the tips of 2 tripods placed at 1 m above the ground, as shown in Figure 3 (left). The participants had to use each method to position 15 anchors while alternating between the target points. The 2 tripods were spaced 5 m apart to emulate a large industrial environment and force the participants to walk around and change position every time. The only exception was with the "create an anchor on the surface" method, as it requires a detected surface to place the anchor. Instead, the targets for it were placed on the ground below the tripods.

In the second part of the experiment, the 4 experimental conditions corresponded to the movement methods and were also counterbalanced with a Latin square. The task was equivalent to the first part, with the only difference that instead of creating new anchors, the users had to move anchors already specified at different points in the scene to the target positions 2.5 m away.

In the first and second part of the study, we collected the responses to the raw NASA-TLX questionnaire [14] for each experimental condition, the total movement of the device as measures of the workload, and task completion times and anchor placement offset as the measures of performance. We dropped the NASA-TLX question regarding the time pressure since it didn't apply. Additionally, we asked questions regarding the fine motor skill demand, the difficulty of learning the method, and the method's overall rating. We chose this approach to keep the questionnaire short but focused on dimensions that seemed most relevant for comparing AR interactions methods.

To evaluate the geometry manipulation system, the task was to fix a misaligned geometric primitive in the 3D scene according to the object in the real world with three repetitions. The participants had to move the three anchors of a virtual box to match the corner points of a real cube (43cm side length) with methods of their choice. In this part of the experiment, we collected subjective data from users about the system and their favourite methods while also tracking each method's usage time.

5.2 Participants

The experiment was performed by twelve Softec AG employees (3 female, 9 male) with an age median of 47. While NASA-TLX has been shown to be valid with sample as small as 6 participants [15], we chose 12 as the sample size common for HCI studies. The sample was chosen to reflect the age distribution of the workforce in an industrial factory. 5 participants stated that they never used an AR application, while only 3 had some experience with an AR-HMD or a complex AR App. The remaining 4 participants reported that they had brief experiences with basic AR apps such as "Pokémon GO".

5.3 Procedure

In the beginning, the participants had to answer questions about their demographics and previous experience with AR, 3D software and smartphone usage. After that, the participants got an explanation of the application and had a chance to explore the complete system. This was done to give everyone a chance to acclimate with the use of an AR application (get used to the correct posture etc.). The first part of the experiment was about anchor creation methods. Before the actual test of each method, the participants were encouraged to experiment with the technique to clear up any questions about the usage and allow for a basic learning phase before the test started. After each experimental condition, the participants had to fill in a NASA-TLX questionnaire and the additional questions to rate the perceived workload and overall rating. Then, the participants had to sit down to fill out the questionnaires and had a chance to take a short break to avoid the influence of fatigue on the results of the study.

After the first part, the experiment continued to evaluate the anchor movement methods similarly to the evaluation of the creation methods.

In the last task the participants subjectively evaluated the geometric primitive editing.

After the tasks were completed, a short informal interview was conducted with the participants, and they provided subjective feedback about the system in general and their preferences.

In total the experiment lasted approx. 90 min for each participant.

5.4 Apparatus

The user study was conducted using the system described in the previous section, executed on a One Plus 6T. The application was extended to incorporate the experimental tasks and to log the relevant data on the device. By placing markers underneath the target points on the tripods, the application was aware of the target points and their positions in the real world. This allowed tracking the distance between the anchors placed by the participants and the actual target point. We also measured the time for the setting of each anchor and the distance travelled by the device as a proxy to the distance covered by the participant. The study was conducted outside, and all devices and relevant surfaces were disinfected to minimise the risk regarding the COVID-19 pandemic.

6 RESULTS

We have collected the data of 720 anchor placements and 720 anchor movements within 96 trials in the user study. We removed the results of all anchor placements with the offset larger than 22 cm since these mainly were unintentional placements (this removes roughly 5% of the 1440 anchors). We chose this cutoff value because nearly all anchors beyond this distance had either extremely short placement times (hinting towards mistakes like hitting a button twice) or were flagged by the instructor due to tracking problems with the device. We have averaged all our raw NASA-TLX scores

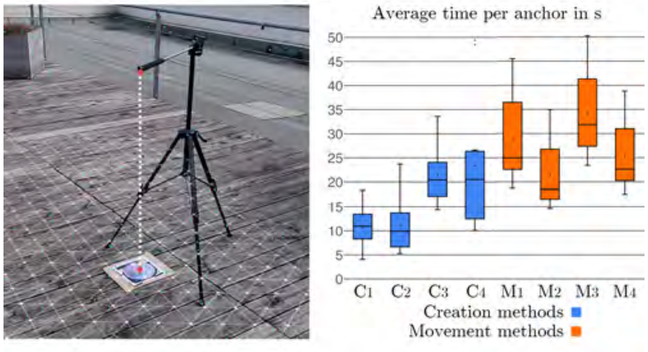

Figure 3: Left: The tracking image at the bottom is used to set the target point (red) in the application to the same position as the tripod grip or tip. The dotted line was added afterwards to better illustrate the relation. Right: The graph shows the Time per Anchor in seconds for the creation methods (blue) and for the movement methods (orange).

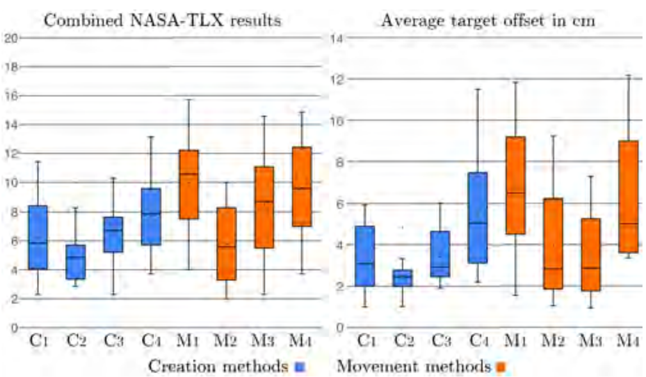

Figure 4: Left: The results of the combined NASA-TLX. Right: Average target offset in cm. The creation methods are blue and the movement methods are orange.

into the Combined NASA-TLX score. For the analysis, we have averaged the numerical values of the anchor placements corresponding to every single trial and used this value for the following statistical test. This allows analysing both measured data and questionnaire results per trial. Due to the small sample size, the results' distribution is non-normal, so we used the Friedmann-test for non-parametric repeated measurements in "SPSS 27". As a post-hoc-test, we used a Dunn-Bonferroni-test. In the Tables 1 and 2, on Figure 3 and 4 and in the following paragraphs we report the test statistic z, the Bonferroni-corrected significance levels p_{bon}, the effect size after Cohen r and the median value M of each method.

To keep the tables and figures concise, we use the condition IDs instead of the full names, as follows: C_1 = Create an anchor at the camera, C_2 = Create an anchor on the surface, C_3 = Create an anchor at the intersection, C_4 = Create an anchor in front of the camera, M_1 = Move an anchor with the camera, M_2 = Move an anchor with the finger, M_3 = Move an anchor with the 3D-movement widget, M_4 = Move an anchor with distance to the camera.

The creation methods' results significantly differ in seven dimensions, but there is no significant difference in the overall rating and the combined NASA-TLX results. In most of the NASA-TLX dimensions, the Bonferroni corrected post-hoc-test does not show significant differences, except for the fine motor skill demand. In this dimension, the method C_1 (M = 7.0) performs significantly better than C_4 (z = -2.846, p_{bon} = .027, r = .43, M = 12.0). The dimension of the average distance to the target differs significantly between C_2 (M = 2.5 cm), and the method C_4 (z = 3.320, p_{bon} = .005, r = .51, M = 5.0 cm). The time per anchor shows a significant split into two groups – one very fast, and the other significantly slower. The participants were able to work significantly faster with the method C_1 (M = 11.0 s) than with the methods C_3 (z = -3.795, p_{bon} = .001, r = .58, M = 20.4 s) and C_4 (z = -3.004, p_{bon} = .016, r = .46, M = 20.5 s). Also the method C_2 (M = 9.9 s) performed quicker than C_3 (z = -3.637, p_{bon} = .002, r = .55, M = 20.4 s) and C_4 (z = 2.846, p_{bon} = .027, r = .43, M = 20.5 s). Regarding the distance travelled between the placement of two anchors the results of C_4 (M = 311 cm) are significantly lower (better) than the results of C_3 (z = -3.637, p_{bon} = .002, r = .55, M = 619 cm) and C_1 (z = -2.846, p_{bon} = .027, r = .43, M = 571cm). The difference between the results of the method C_2 (M = 496 cm) and the method C_3 (z = -3.162, p_{bon} = .009, r = .48, M = 619 cm) in this respect is also significant.

Concerning the 4 movement methods, the post-hoc-test shows significant differences in all but one dimension (physical demand) in the results. The combined NASA-TLX results show that the method M_2 (M = 4.6) performed significantly better than the methods M_4 (z = 3.637, p_{bon} = .002, r = .55, M = 8.6) and M_1 (z = -4.111, p_{bon} < .001, r = .63, M = 9.6). The differences for these pairs are also significant for the overall rating dimension: M_2 (M = 3.0) to M_4 (z = 4.032, p_{bon} < .001, r = .61, M = 9.0); M_2 to M_1 (z = -3.162, p_{bon} = .009, r = .48, M = 7.0). The achieved distance to the target shows significant differences between the method M_3 (M = 2.9 cm), and M_1 (z = 2.846, p_{bon} = .027, r = 0.43, M = 6.5 cm) respectively to M_4 (z = -2.846, p_{bon} = .027, r = .43, M = 5.0 cm). The participants finished the task significantly faster with M_2 (M = 18.5 s) than with M_3 (z = -4.427, p_{bon} < .001, r = .67, M = 31.9 s), and M_1 (z = 2.846, p_{bon} = 0.027, r = .43, M = 25.0 s). Additionally, the time difference between the method M_4 (M = 22.7 s) and the method M_3 (z = 2.846, p_{bon} = .027, r = .43, M = 31.9 s) is significant. Lastly the method M_2 (M = 341 cm) has a significantly shorter distance travelled per anchor than M_1 (z = 3.637, p_{bon} = .002, r = .55, M = 645 cm) and M_3 (z = -3.162, p_{bon} = .009, r = .48, M = 571 cm).

The box fitting task results show the percentage of time a movement method was selected by the participants while working on the task. In 42.7% of the time, the participants had no method chosen and spent the time checking their progress or planning the next step. A significant part of the rest of the time the participants used the method M_2 (36,0%) followed by the method M_3 (15.6%); M_1 (4.2%) and M_4 (1.6%) achieved only a short usage time.

7 DISCUSSION

In the following, we will go over each method and interpret the results while also adding some of the qualitative feedback given by the participants and our observations.

	Median				Friedmann	post-hoc p-values					
Dimension	C_1	C_2	C_3	C_4	p-value	C_1:C_2	C_1:C_3	C_1:C_4	C_2:C_3	C_2:C_4	C_3:C_4
Mental	4.0	4.0	5.0	8.0	**.029**	1	1	.197	1	.106	1
Physical	5.0	5.0	6.0	4.0	.284	1	1	1	1	1	1
Fine motor skill	7.0	6.0	13.0	12.0	**.003**	1	.068	**.027**	.161	.068	1
Performance	6.0	5.0	5.0	6.0	.560	1	1	1	1	1	1
Effort	5.0	4.0	6.0	8.0	**.045**	1	1	.239	1	.289	.131
Frustration	4.0	3.0	4.0	6.0	.070	1	1	1	1	1	1
Difficulty of learning	4.0	4.0	4.0	6.0	**.011**	1	1	.289	1	.086	.161
Overall rating	4.0	3.0	5.0	5.0	.423	1	1	1	1	1	1
Combined NASA-TLX	4.9	3.9	5.7	6.9	.053	1	1	1	1	1	1
Target offset in cm	3.1	2.5	2.9	5.0	**.008**	.928	1	.347	1	**.005**	.106
Time in s	11.0	9.9	20.4	20.5	**<.001**	1	**.001**	**.016**	**.002**	**.027**	1
Travelled dist. in cm	571	496	619	311	**<.001**	.106	1	**.027**	**.009**	1	**.002**

Table 1: The results for all creation methods.

	Median				Friedmann	post-hoc p-values					
Dimension	M_1	M_2	M_3	M_4	p-value	M_1:M_2	M_1:M_3	M_1:M_4	M_2:M_3	M_2:M_4	M_3:M_4
Mental	12.0	5.0	9.0	12.0	**<.001**	**.001**	1	1	**.027**	**.007**	1
Physical	12.0	4.0	7.0	6.0	**.035**	.086	1	1	.239	.414	1
Fine motor skill	14.0	8.0	12.0	11.0	**.002**	**.012**	1	1	.106	**.012**	1
Performance	7.0	5.0	5.0	8.0	**<.001**	.086	**.012**	1	1	.055	**.007**
Effort	12.0	5.0	8.0	11.0	**.001**	**.002**	.492	1	.347	**.043**	1
Frustration	8.0	4.0	6.0	10.0	**.006**	**.034**	1	1	.492	**.034**	1
Difficulty of learning	7.0	4.0	6.0	8.0	**.008**	**.043**	1	1	.131	.055	1
Overall rating	7.0	3.0	6.0	9.0	**<.001**	**.009**	1	1	.289	**<.001**	0.239
Combined NASA-TLX	9.6	4.6	7.7	8.6	**<.001**	**<.001**	.492	1	.106	**.002**	1
Target offset in cm	6.5	2.8	2.9	5.0	**.004**	.161	**.027**	1	1	.161	**.027**
Time in s	25.0	18.5	31.9	22.7	**<.001**	**.027**	.683	1	**<.001**	.683	**.027**
Travelled dist. in cm	645	341	571	381	**.001**	**.002**	1	.161	**.009**	.928	.492

Table 2: The results for all movement methods.

Create an anchor at the camera (C_1): The participants reported low physical, mental, fine motor skill demand and a low effort, achieving a low time per anchor. Still, the offset to the target is only mediocre compared to the other methods. This can be attributed to most participants' hesitation to put the device as close as possible to the target point before placing the anchor. This hesitation faded during the test reducing the distance. However, this learning effect only started under test conditions since the users were focused on precision only in the trial and not in the try out phase before the test.

Create an anchor on the surface (C_2): This method achieved the best results in the NASA-TLX test and the lowest time per anchor combined with the lowest offset to the target. However, the technique is not fully comparable to the others because it can only place an anchor on a defined or detected surface, omitting one degree of freedom.

Create an anchor at the intersection (C_3): Surprisingly, the participants reported the same difficulty in learning and a similar mental load while using this method as using the previous two, even though this method requires two inputs. In addition, the participants moved a lot around during the test, even though the technique would allow for far less movement. The NASA-TLX results are lower than for the other two methods, probably caused by the high fine motor skill rating.

Create an anchor in front of the camera (C_4): This method achieved the worst results of the 4 creation methods in nearly every dimension. Based on the observations during the test, we attribute this to the small device movements that occur by touching and releasing the screen, similar to the "Heisenberg Effect of Spatial Interaction" [29]. In addition, some of the users reported that they had difficulties splitting up their attention between controlling the height and the position of the anchor on the horizontal plane.

Move an anchor with the camera (M_1): Despite multiple participants noting that they had fun with this method, it achieved the worst results of all 4 movement methods. This seems especially interesting since similar implementations of this method achieved good results in other test configurations [22, 26]. A relevant difference to these studies could be the scale, as the participants had to move the anchor over distances of more than 1*m* from a random starting point. This also led to the problem that some participants activated the interaction before finding the anchor with the screen, which means the anchor moved with the device while staying permanently off the screen. This could be avoided by disabling the method when the selected anchor is not visible.

Move an anchor with the finger (M_2): This method achieved the best NASA-TLX result of the 4 movement methods and the lowest offset to the target while being the fastest. It also achieved the highest usage time in the box fitting test, and multiple participants chose it as their favourite method in the post-test questionnaire.

Move an anchor with the 3D-movement widget (M_3): This method achieved a similar target offset as the method M_2, however it took the participants 72% longer to position the anchors, and the NASA-TLX results are lower. In addition, some participants initially faced difficulties understanding and using the widget, especially when moving the anchor diagonally to the axes.

Move an anchor with distance to the camera (M_4): This method performed like the method C_4 for similar reasons. Even though the functionality differs between the methods, based on the user feedback, the focus split between control of the anchor position and distance to the anchor seems to be the reason for the high metal demand.

Box fitting task: This task showed that most participants favoured the methods M_2 and M_3 over the others for the required small position changes. In addition, some users had problems understanding how the 3 anchors control the volume. After further explanations, however, they were able to manipulate the anchors to approximate the cube.

8 LIMITATIONS

The study's design makes a general comparison of the tested methods possible, which leads to the recommendations in the next paragraph. However, it also has some limitations.

The generalised test setup allows for comparison while suppressing the strengths of some methods, e.g. M_3 enables the users to place anchors at hard-to-reach points in the distance, which was not part of the test and is not possible with some other methods. In a real industrial environment, it might be necessary to approximate machines bigger than 5*m*, and for such use cases, the method selection might be different. However, the results of this study can still be used to make an informed design decision for other environments. For example, if a method achieved a low precision here, it will most likely perform similar or worse if the users can not get close to the target position.

The user study has been executed on a smartphone with a 6.4″ screen, while the system can also be used with a tablet PC, and then, the larger screen might change the choice of interaction method.

While having provided enough time for the participants to get used to the system and the interaction methods, they should still be considered novice users. This does not match the industrial environment well, where prospective users are supposed to get the corresponding training before using the system.

The decision to start the test with the creation methods was made to prepare the participants for the slightly more complex test setup of the movement methods. However, this possibly introduced an ordering effect, and some participants still faced problems with the general flow of the movement test, which could explain the overall lower results of the movement methods compared to the creation methods.

With larger sample size we would observe narrower confidence intervals and more granular distinctions between the methods for the collected variables. However, considering mostly quantitative data, and high reliability of NASA-TLX [15], this study should provide a valid overview and comparison of the different interaction methods, and is suitable to select methods for further development.

In the analysis we were interested not merely in testing if there are differences between the methods, but in identifying the types of differences, thus we performed statistical tests for all dependent variables, as reported in the tables. However, to avoid p-value inflation we have used Bonferroni corrected Fridman omnibus tests and Dunn-Bonferroni post hoc tests. We report in the text and draw conclusions according to their results.

The target offset values should not be taken as an absolute value as tracking errors of ARCore impact the actual results of the test.

Anchor size could influence the behaviour of the user. For example, a smaller anchor could encourage a more exact placement.

9 DESIGN RECOMMENDATIONS

- Depending on the complexity and type of the task, users should have different interaction methods to choose from. One interaction technique cannot fit all use cases. This requires some form of training for users to make an informed decision based on the situation.
- According to the qualitative participant feedback, the combination of interactions that do not interfere with each other could be beneficial, for example, allowing the "move with the finger" method while the widget of "move with the 3D-movement widget" is active.
- When combining device-based and touch-based inputs into one multimodal interaction, the jitter and disruption of the device introduced by the touch inputs should be considered.
- Based on the better results for touch-based interactions, those methods should be preferred over other options.
- Splitting the user's attention between two points while controlling both points at once should be avoided (e.g. drop shadow and anchor in the "move with distance to the camera" method).

10 CONCLUSION

This paper has designed 3 interaction methods for anchor placement and 2 methods for anchor movement in a mobile AR application and evaluated them together with 3 techniques from the literature. In contrast to the previous work, we designed our interactions to move single anchors instead of complex objects. The interaction techniques were implemented as part of a mobile AR application for visualising data in an industrial environment. The interaction techniques allow users to place the visualisations efficiently and flexibly create geometric primitives in an AR environment. The user study showed that touch-based methods significantly outperformed the camera and multimodal methods, which provided the foundation for the corresponding design recommendations.

REFERENCES

[1] Imran Ashraf, Soojung Hur, and Yongwan Park. 2018. MagIO: Magnetic Field Strength Based Indoor- Outdoor Detection with a Commercial Smartphone. *Micromachines* 9 (10 2018), 534. https://doi.org/10.3390/mi9100534

[2] Benjamin Bach, Ronell Sicat, Hanspeter Pfister, and Aaron Quigley. 2017. Drawing into the AR-CANVAS: Designing Embedded Visualizations for Augmented Reality. In *Workshop on Immersive Analytics, IEEE Vis*.

[3] C. Diaz, M. Walker, D. A. Szafir, and D. Szafir. 2017. Designing for Depth Perceptions in Augmented Reality. In *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 111–122. https://doi.org/10.1109/ISMAR.2017.28

[4] Ze Dong, Thammathip Piumsomboon, Jingjing Zhang, Adrian Clark, Huidong Bai, and Rob Lindeman. 2020. A Comparison of Surface and Motion User-Defined Gestures for Mobile Augmented Reality. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '20)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/3334480.3382883

[5] Ruofei Du, Eric Turner, Maksym Dzitsiuk, Luca Prasso, Ivo Duarte, Jason Dourgarian, Joao Afonso, Jose Pascoal, Josh Gladstone, Nuno Cruces, Shahram Izadi, Adarsh Kowdle, Konstantine Tsotsos, and David Kim. 2020. *DepthLab: Real-Time 3D Interaction with Depth Maps for Mobile Augmented Reality*. Association for Computing Machinery, New York, NY, USA, 829–843. https://doi.org/10.1145/3379337.3415881

[6] Chen Feng and Vineet Kamat. 2012. AUGMENTED REALITY MARKERS AS SPATIAL INDICES FOR INDOOR MOBILE AECFM APPLICATIONS. https://doi.org/10.13140/2.1.4484.4166

[7] Jan Fischer, Benjamin Huhle, and Andreas Schilling. 2007. Using Time-of-Flight Range Data for Occlusion Handling in Augmented Reality. In *Eurographics Symposium on Virtual Environments*, Bernd Froehlich, Roland Blach, and Robert van Liere (Eds.). The Eurographics Association.

[8] Borko Furht (Ed.). 2008. *Multimodal Interfaces*. Springer US, Boston, MA, 651–652. https://doi.org/$10.1007/978-0-387-78414-4_159$

[9] J. Gimeno, S. Casas, C. Portalés, and M. Fernádez. 2018. Addressing the Occlusion Problem in Augmented Reality Environments with Phantom Hollow Objects. In *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. 21–24. https://doi.org/10.1109/ISMAR-Adjunct.2018.00024

[10] E. S. Goh, M. S. Sunar, and A. W. Ismail. 2019. 3D Object Manipulation Techniques in Handheld Mobile Augmented Reality Interface: A Review. *IEEE Access* 7 (2019), 40581–40601. https://doi.org/10.1109/ACCESS.2019.2906394

[11] Google LLC. [n.d.]. (ARCore) Quickstart for Android. https://developers.google.com/ar/develop/unity/quickstart-android accessed: 26.01.2021.

[12] Google LLC. [n.d.]. Depth API overview for Android. https://developers.google.com/ar/develop/java/depth/overview accessed: 26.01.2021.

[13] Mark Hancock, Thomas Ten Cate, and Sheelagh Carpendale. 2009. Sticky tools: full 6DOF force-based interaction for multi-touch tables. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. 133–140.

[14] Sandra G. Hart. 2006. Nasa-Task Load Index (NASA-TLX); 20 Years Later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 50, 9 (2006), 904–908. https://doi.org/10.1177/154193120605000909 arXiv:https://doi.org/10.1177/154193120605000909

[15] Susan G. Hill, Helene P. Iavecchia, James C. Byers, Jr. Alvah C. Bittner, Allen L. Zaklade, and Richard E. Christ. 1992. Comparison of Four Subjective Workload Rating Scales. *Human Factors* 34, 4 (1992), 429–439. https://doi.org/10.1177/001872089203400405 arXiv:https://doi.org/10.1177/001872089203400405

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

[17] Ajune Ismail and Mohd Shahrizal Sunar. 2015. Multimodal Fusion: Gesture and Speech Input in Augmented Reality Environment. *Advances in Intelligent Systems and Computing* 331 (01 2015), 245–254. https://doi.org/$10.1007/978-3-319-13153-5_24$

[18] Minkyung Lee, Mark Billinghurst, Woonhyuk Baek, Richard Green, and Woontack Woo. 2013. A usability study of multimodal input in an augmented reality environment. *Virtual Reality* 17 (11 2013). https://doi.org/10.1007/s10055-013-0230-0

[19] Jingbo Liu, Oscar Kin-Chung Au, Hongbo Fu, and Chiew-Lan Tai. 2012. Two-finger gestures for 6DOF manipulation of 3D objects. In *Computer Graphics Forum*, Vol. 31. Wiley Online Library, 2047–2055.

[20] Anthony Martinet, Gery Casiez, and Laurent Grisoni. 2011. Integrality and separability of multitouch interaction techniques in 3D manipulation tasks. *IEEE transactions on visualization and computer graphics* 18, 3 (2011), 369–380.

[21] Anthony Martinet, Géry Casiez, and Laurent Grisoni. 2011. Integrality and Separability of Multitouch Interaction Techniques in 3D Manipulation Tasks. *IEEE transactions on visualization and computer graphics* 18 (07 2011), 369–80. https://doi.org/10.1109/TVCG.2011.129

[22] Annette Mossel, Benjamin Venditti, and Hannes Kaufmann. 2013. 3DTouch and HOMER-S: Intuitive Manipulation Techniques for One-Handed Handheld Augmented Reality *(VRIC '13)*. Association for Computing Machinery, New York, NY, USA, Article 12, 10 pages. https://doi.org/10.1145/2466816.2466829

[23] Thammathip Piumsomboon, David Altimira, Hyungon Kim, Adrian Clark, Gun Lee, and Mark Billinghurst. 2014. Grasp-Shell vs gesture-speech: A comparison of direct and indirect natural interaction techniques in augmented reality. https://doi.org/10.1109/ISMAR.2014.6948411

[24] Ali Samini and Karljohan Lundin Palmerius. 2016. A Study on Improving Close and Distant Device Movement Pose Manipulation for Hand-Held Augmented Reality. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology* (Munich, Germany) *(VRST '16)*. Association for Computing Machinery, New York, NY, USA, 121–128. https://doi.org/10.1145/2993369.2993380

[25] Aditya Sankar and Steven Seitz. 2016. In situ CAD capture. 233–243. https://doi.org/10.1145/2935334.2935337

[26] Tomohiro Tanikawa, Hidenori Uzuka, Takuji Narumi, and Michitaka Hirose. 2015. Integrated View-Input Ar Interaction for Virtual Object Manipulation Using Tablets and Smartphones. In *Proceedings of the 12th International Conference on Advances in Computer Entertainment Technology* (Iskandar, Malaysia) *(ACE '15)*. Association for Computing Machinery, New York, NY, USA, Article 7, 8 pages. https://doi.org/10.1145/2832932.2832956

[27] Wang, Wei, Wang, Fei, Song, Wei, and Su, Shun. 2020. Application of Augmented Reality (AR) Technologies in inhouse Logistics. *E3S Web Conf.* 145 (2020), 02018. https://doi.org/10.1051/e3sconf/202014502018

[28] W. Willett, Y. Jansen, and P. Dragicevic. 2017. Embedded Data Representations. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 461–470. https://doi.org/10.1109/TVCG.2016.2598608

[29] Dennis Wolf, Jan Gugenheimer, Marco Combosch, and Enrico Rukzio. 2020. Understanding the Heisenberg Effect of Spatial Interaction: A Selection Induced Error for Spatially Tracked Input Devices. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3313831.3376876