

Pocket6: A 6DoF Controller Based On A Simple Smartphone Application

Teo Babic
BMW Group
Munich, Germany
teo.babic@bmw.de

Harald Reiterer
HCI Group, University of Konstanz
Konstanz, Germany
harald.reiterer@uni-konstanz.de

Michael Haller
Media Interaction Lab, University of
Applied Sciences Upper Austria
Hagenberg, Austria
michael.haller@fh-hagenberg.at

ABSTRACT

We propose, implement and evaluate the use of a smartphone application for real-time six-degrees-of-freedom user input. We show that our app-based approach achieves high accuracy and goes head-to-head with expensive externally tracked controllers. The strength of our application is that it is simple to implement and is highly accessible — requiring only an off-the-shelf smartphone, without any external trackers, markers, or wearables. Due to its inside-out tracking and its automatic remapping algorithm, users can comfortably perform subtle 3D inputs everywhere (world-scale), without any spatial or postural limitations. For example, they can interact while standing, sitting or while having their hands down by their sides. Finally, we also show its use in a wide range of applications for 2D and 3D object manipulation, thereby demonstrating its suitability for diverse real-world scenarios.

CCS CONCEPTS

• Human-centered computing → Mixed / augmented reality; Pointing devices; Mobile phones;

KEYWORDS

Smartphone; 3D interaction; augmented reality; 6DoF input

ACM Reference Format:

Teo Babic, Harald Reiterer, and Michael Haller. 2018. Pocket6: A 6DoF Controller Based On A Simple Smartphone Application. In *Symposium on Spatial User Interaction (SUI '18)*, October 13–14, 2018, Berlin, Germany. ACM, Berlin, Germany, Article 4, 9 pages. <https://doi.org/10.1145/3267782.3267785>

1 INTRODUCTION

Externally-tracked six-degrees-of-freedom (6DoF) handheld input controllers set the standard for 3D interaction. Users typically find them very intuitive to use, since their translation/rotation based movement detection allows them to be used as "virtual hands" to locate, grab, and manipulate 3D objects — as is done in the real-world. Unfortunately, when external tracking is not possible (i.e.

due to the lack of infrastructure, on-the-go usage, etc.), the interaction possibilities are drastically reduced and become highly device-dependent. In such cases, mobile handheld controllers (popular with recent mobile VR/AR headsets), or wearable devices with hand-motion tracking capabilities may be used. However, these require specialized hardware and/or configuration which prohibits spontaneous usage. Additionally, many are also incapable of 6DoF tracking. Previous research works have experimented with using the most convenient and available end-user device, a personal smartphone, as a means to perform 2D and 3D interactions. However, they have so far either required external hardware (e.g. cameras, trackers, printed markers) to make their phones spatially-aware, or have proposed interaction techniques based only on 3DoF rather than 6DoF input. As such, we wished to explore the possibility of a smartphone application that can overcome these challenges.

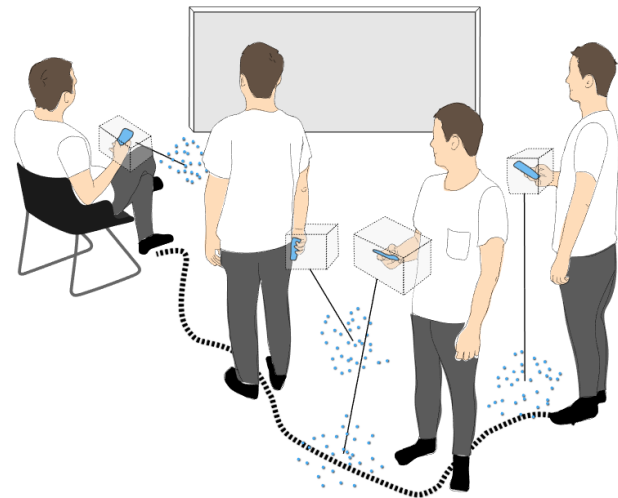


Figure 1: Pocket6 is an AR application for 6DoF user inputs from a subtle control space which automatically recalibrates based on user's orientation, position, and posture changes.

In this work, we present *Pocket6*, a smartphone application that uses the AR tracking capabilities of modern smartphones to enable 6DoF input without requiring external hardware. We show that users can use Pocket6 by performing accurate subtle gestures within a control space of only $16 \times 9 \times 16$ cm. With our novel approach, users can freely change their position, orientation, or posture, due to our auto-calibration feature for the control space, cf

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SUI '18, October 13–14, 2018, Berlin, Germany

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5708-1/18/10...\$15.00

<https://doi.org/10.1145/3267782.3267785>

Figure 1. We extensively evaluate the performance of our approach and compare it against a high-end VR controller. Furthermore, we evaluate the impact body postures (i.e. *standing*, *sitting*, and *hand-down*) have on user-performance with the application. We conclude by demonstrating applications and exposing scenarios in which our system will be beneficial.

2 RELATED WORK

Novel input devices (e.g. VR controller wands, etc.) have become increasingly more popular and common in the consumer market. Besides commercial 6DoF devices, smartphones and other objects have been used in conjunction with reflective markers and external reference cameras to serve as controllers (e.g. OptiTrack [41]). With such systems, users could spatially translate the smartphone to navigate pan-and-zoom interfaces on smartphones [6, 43, 55], and other screens [7, 40]. Other works have focused on their use for 3D object manipulation [6, 8], proximity-aware interfaces [31, 36] as well as for media transfer between devices [54].

In most cases, researchers have indicated that using external tracking has considerable disadvantages [11, 28, 33]. Although sensor capabilities will improve over time, problems with occlusion will persist and will continue to require users to perform large, explicit hand/arm gestures in front of their bodies [33]. These are more tiring, socially-conspicuous, and difficult to perform within a small physical space [25, 27]. Generally, exclusive controllers have been found to come with a lot of shortcomings, regarding range limitations, acquisition times, controller size, connector cables, energy requirements, and manufacturing costs [28].

2.1 Interacting with the smartphone

Smartphones can be a powerful alternative to exclusive controllers. With the Inertial Measurement Unit (IMU), phone-orientation (3DoF) has been used to point at distant screens [11, 15, 44, 50] and in VR [16, 32] via ray-casting. The IMU has furthermore been used for the selection and manipulation of objects in 3D environments via 2D rotational planes [19, 29, 45, 53, 57], for enabling *throw*- or *swing* gestures to transfer media between devices [14, 42], and for uni-stroke letter [1] or symbol [30, 63] recognition in combination with acceleration data.

Many studies have shown that using the phone's movements, combined with touch, can outperform the use of many other options (i.e. touch-only devices, mouse, Wii remotes, 6DoF wand) in 3D object translation/rotation tasks [28, 29, 44, 56]. However, this requires users to calibrate their smartphone's IMU to determine its correct orientation [15, 16, 50] each time they intend to use it. Moreover, calibration may need to be repeated, user-initiated or partly-automated during interaction to maintain input accuracy. These works demonstrate that IMUs are generally sufficient for the discrete detection of motion, but are inadequate for precise and continuous position tracking necessary for 6DoF tracking. Since IMUs continually integrate acceleration with respect to time to calculate velocity and position (dead reckoning), any measurement errors, however small, accumulate over time [18, 51], leading to "drift": an ever-increasing difference between where the device is thought to be located and its actual location.

2.2 Smartphone Camera Enabled Tracking

Rekimoto [46] proposed an approach for simultaneously determining a smartphone's position and orientation using printed 2D matrix markers (square shaped barcodes on paper) attached to objects in the environment. With the appropriate phone application (e.g. ARToolKit [3, 24, 38]) the phone's camera could seek out and identify external markers to estimate the camera's relative position and orientation. Based on this technology, researchers proposed the "Sweep" and "Point-and-Shoot" techniques for relative cursor control and selection of targets on a large screen [5, 47, 58]. They furthermore demonstrated techniques for 3D object manipulation [23], 3D mesh editing in 6DoF [22], bi-manual interaction [20, 61] and map interactions [47]. They also combined phone movement with touch input [37, 39] to manipulate AR object displayed on the phone's screen. Later, researchers investigated 2D interactions based on optical-flow analysis [9, 10, 62], making markers obsolete. Wang et al. [62] proposed an interesting concept that enabled 2D gestures for phone control that could be used outdoors. Now, modern smartphones can use more advanced computer-vision methods (e.g. dense SLAM [49] or similar [17, 21]) to detect the precise surface geometry of an unknown environment and later use it to estimate its own position and rotation. Such solutions empower us to progress beyond local 6DoF tracking.

3 POCKET6

The overall concept of Pocket6 is based on Apple's Augmented Reality toolkit (ARKit, v1.5), which was launched in 2017. ARKit is an inside-out mobile tracking software which employs a technique known as sensor fusion. It uses data from the smartphone's existing camera and IMU sensors to calculate where the device is in space and translates the physical location and movement into a movement within a virtual environment. Instead of anchoring virtual objects in real-world coordinates, we use ARKit for anchoring a virtual 3D control space, in which the phone's location is tracked, cf. Figure 2b. This setup allows for 2D and 3D input to external applications cf. Figure 2f,g. We tested Pocket6 on an iPhone X, but it could also be installed on other ARKit-supported devices, such as an iPhone SE, 6s, etc [2]. Other similar toolkits would support the same implementation, such as Google's ARCore or Vuforia. Since Pocket6 was used as an input application only, we implemented a Windows program which could display output applications on an external display. This application communicated with Pocket6 over WiFi at 60 Hz.

3.1 Control Space Auto-Calibration

One of the primary goals of the implementation was to eliminate the need for users to initiate calibration procedures. As such, we implemented an auto-calibration algorithm which re-calibrates the control space position and rotation whenever users: (a) open the application for the first time, (b) move their hand to a different position, or (c) change their posture or walk around, cf. Figure 2d. In each case, the control space *follows* the users' motion, consequently enclosing the users' hand at all times. This is done through the following three steps.

- (1) *Control Space Anchoring*: Firstly, based on the 3D Cartesian coordinate system in real-world metric units (cm) defined

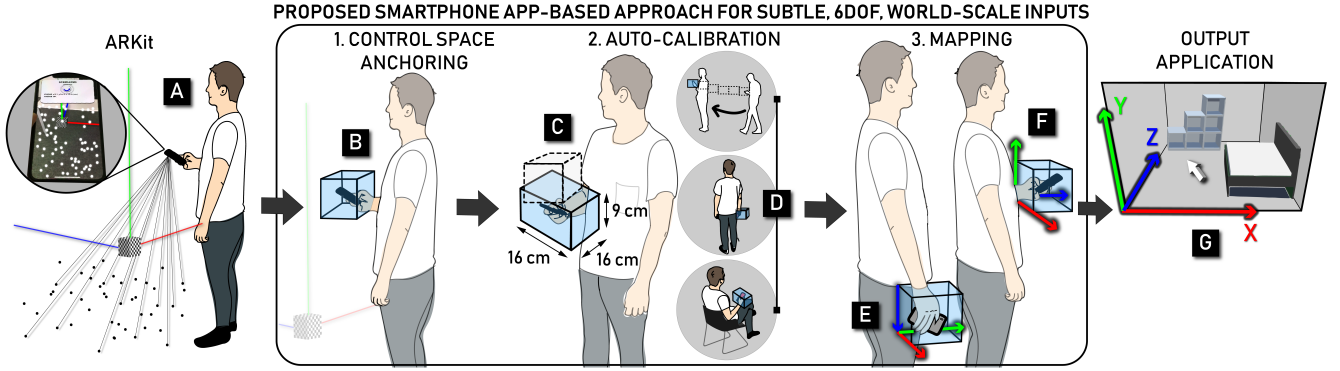


Figure 2: A user interacts within a virtual 3D control space (visualized as a bounding box). The coordinates of the real interaction space are mapped automatically to the digital screen. The movement of the cursor corresponds with the smartphone's motions.

by ARKit (Figure 2a), we align the geometric center of the control space with the smartphone's position (Figure 2b), and align their (z-axis) rotations (Figure 2c). The boundaries of the control space were determined by the results of an empirical study performed during its development, which suggested that the control space should reflect the aspect ratio of the output device at the ratio of 1 cm:120 px (control space:output device). In our setup, a 1920×1080 px screen maps to a control space of 16×9 cm (along the x- and y-axis). In 3D applications, a depth of 1920 px was established, which mapped to a 16 cm depth for the control space (along the z-axis), cf. Figure 2c.

- (2) *Auto-Calibration*: Whenever the phone's position exits the boundaries of the control space, cf. Figure 2c, it is assumed that the user has changed his/her position, triggering a recalibration step. Since our approach also allows users to interact with the controlling hand pointing in a downward position, the axes of our control space must adapt accordingly [33], cf. Figure 2e. This occurs automatically once an upside-down orientation is detected by the phone (i.e. the phone's x-axis rotation is between 130 and 230).
- (3) *Mapping*: Finally, we normalize the smartphone's position within the control space boundaries and forward its data (i.e. position, rotation, touchscreen events) to an application, e.g. for moving a 3D cursor, etc. (cf. Figure 2g).

To minimize both jitter and latency, we used the 1 filter [12]. No other forms of signal processing was used.

4 EVALUATION

We conducted an empirical study with three experiments to explore the benefits and limitations of our proposed app-based tracking approach. We compared Pocket6 to a high-end, state-of-the-art, VR controller. Furthermore, we investigated how well can participants perform input from different body postures. All three experiments were conducted with the same techniques, participants, and apparatus.

Four different conditions (see Figure 3) were tested to compare the performance, accuracy as well as subjective feedback of our proposed condition:

- *In-Front*: When using the *In-Front* condition, participants were standing while naturally holding the smartphone *in-front* of their torso.
- *Sitting*: Participants were *sitting* on a chair and their elbow was resting on the chair's armrest, while holding the smartphone above their waist.
- *Hands-Down*: In this condition, participants were standing while holding the smartphone in a *hands-down* posture.
- *Baseline*: In the baseline condition, participants used an externally-tracked VR controller, participants were standing while holding the VR controller in-front of their torso.

4.1 Apparatus

The study was conducted in a quiet room, where the participants stood/sat two metres away from the display, a 1920×1080 pixel 32" Samsung TV, which was showing all the tasks. For the *Baseline* condition, we used a HP Microsoft Mixed Reality headset and one of its Windows Mixed Reality motion controllers. Since the controller tracking cameras are embedded in the headset itself, we mounted the headset on a tripod and placed it in front of the participant, so that it could easily capture all their hand motions (approx. at a distance of 0.5 to 0.7 m). The height of the headset was adjusted and optimized for each participants' chest height. The raw data received from the headset controller and Pocket6 were processed through exactly the same signal filtering, control space remapping and mapping algorithms.

4.2 Participants

In total, 12 paid volunteers (5 females, 7 males) from the local entity participated whose age ranged from 20 to 42 years ($M = 31.5$, $SD = 5.17$). They were all right-handed, none of them had previous experiences with 6DoF input controllers, and all of them used their smartphones on a daily basis.

4.3 Study Design

A repeated measures within-subject design was used. We investigated the participants' performance in a 2D tracking task (1), 2D Fitts' law task (2), and 3D placement task (3). These three tasks were built on each other with an increasing input complexity and were performed in sequence. Thus, participants incrementally practiced while they progressed from easy to more complex tasks. Each participant was welcomed and introduced to the experiment procedure. After filling out a background questionnaire, they were given time to practice with the conditions until they felt comfortable with the system. Performance data was captured through computer logs and subjective feedback data was collected by an exit questionnaire.

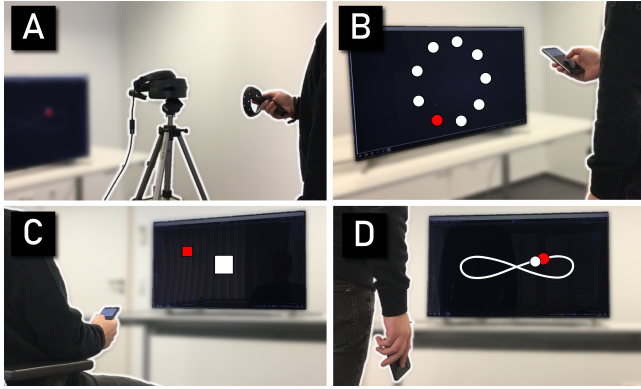


Figure 3: Study tasks and conditions: Baseline condition (a), 2D pointing task with the *In-Front* condition (b), 3D placement task with the *Sitting* condition (c) and the tracing task with the *Hands-Down* condition.

4.4 Experiment 1: 2D Tracking

The purpose of the first experiment was to use a tracking task to determine if different input conditions have an influence on the users' ability to precisely guide their hand in mid-air. In fact, we wanted to know how accurately participants can continuously follow a moving target with the cursor - even when the target did fast directional changes.

Our task was based on the ideas of [13, 52], where participants had to trace a target moving on a pre-defined path. Similarly as in [13], we choose an narrow 720×270 px eight-shaped path (∞), and used a target movement speed of 2π seconds, which results in a continuous and fluid motion of the hand and defines the duration of one trial (lap). We chose an eight-shaped path since it is more complex compared to a circular or elliptical path and required a more fine-granular input from the participants. Each trial started/ended when the target crossed the bottom part of the left loop. From there the target was moving clockwise along the left loop and counter-clockwise along the right loop. Both the participant's-controlled cursor as well as the target had the radius size of 50 px.

Each participant had to perform seven trials for each of the four conditions. They started by two practice trials, followed by five (measured) trials. In summary, each participant performed 20 trials.

During the experiment, we logged the Cartesian distance in pixels (px) between the path and cursor at a rate of 60 Hz.

4.4.1 Results. Figure 4 depicts all traces for each condition of all participants. In the *Baseline* condition, achieved an average error of 68.60 px ($SD = 15.65$), for *In-Front* 73.10 px ($SD = 20.40$), for *Sitting* 73.97 px ($SD = 21.26$), and finally for the *Hands-Down* condition 90.10 px ($SD = 29.91$).

A one-way repeated measures ANOVA ($\alpha = .05$) showed significant differences between the four conditions ($F_{3,33} = 8.152, p < 0.001$). As the collected data did not violate the assumption of sphericity, no corrections were necessary. The post-hoc pairwise comparison using Bonferroni corrections showed that the *Hands-Down* condition was significantly more inaccurate (on average for 16 to 21 px), compared to all other conditions (*Baseline* $p = 0.001$, *In-Front* $p = 0.014$, *Sitting* $p = 0.002$). *Baseline*, *In-Front* and *Sitting* conditions did not differ significantly, on average for less than 6 px.

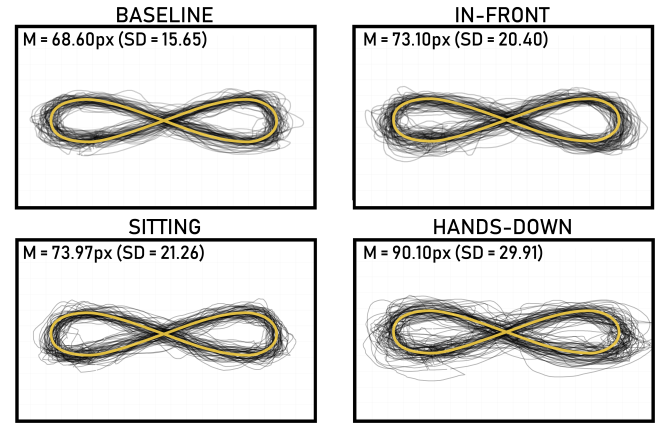


Figure 4: Cursor traces of all measured tracking tasks, for each of the four conditions. The yellow line indicates the target's eight-shaped path.

We see, that the Pocket6 conditions *In-Front* and *Sitting* allowed participants to perform continuous hand motions as fine-granular and precise as an externally tracked controller, the *Baseline* condition. Furthermore, we can see that in the *Hands-Down* condition participants performed less accurate than in all other conditions.

4.5 Experiment 2: 2D Pointing

The goal of the second experiment was to evaluate 2D pointing and clicking performance. This experiment was based on the ideas of [33, 59], who used a 2D Fitts' law task [26, 48], which is accessible on [35]. In this experiment, participants had to point and click a set of circular targets displayed on the screen. A trial was successful once the first *click-down* and *click-up* events occurred inside the target boundaries. Each target had to be successfully selected to continue to the next trial. We compared two amplitudes (400 and 800 px) and three target widths (50, 100, and 200 px) creating an Index of Difficulty (ID) range of 1.6 to 4.1 bits. For each of the four conditions, participants had to finish a block of practice trials with 3 targets, followed by a block of randomized measured trials

with 9 targets. Each block contained all combinations of amplitudes and widths. In total, each participant generated 216 data entries: 4 conditions \times 2 amplitudes \times 3 widths \times 9 target selections.

4.5.1 Results. We analyzed the main effects of our conditions on the traditional measures of throughput, error rate, and movement time. We used repeated measures ANOVA ($\alpha = .05$) and pairwise tests with Bonferroni corrections for the post-hoc analysis.

The *Baseline* condition had a throughput of 1.86 bps ($SD = 0.59$), *In-Front* 1.82 bps ($SD = 0.55$), *Sitting* 1.83 bps ($SD = 0.52$) and *Hands-Down* 1.36 bps ($SD = 0.50$). A one-way ANOVA showed a significant difference for the four conditions ($F_{3,15} = 8.00, p < 0.006$). The assumption of sphericity was violated, so the Greenhouse-Geisser corrected values are reported. The post-hoc test showed that *Hands-Down* had a significantly lower throughput (on average 26%) compared to all other conditions: *Baseline* ($p = 0.027$), *In-Front* ($p = 0.01$), and *Sitting* ($p < 0.001$). Other pairs did not differ significantly, their throughput differed on average for less than 2%.

The average error-rate for *Sitting* was 8.3% ($SD = 10.59$), *In-Front* 11.26% ($SD = 13.54$), *Baseline* 12.19% ($SD = 12.98$), and *Hands-Down* 18.05% ($SD = 15.31$). A one-way ANOVA showed a significant difference between the conditions ($F_{3,33} = 9.720, p < 0.001$). The assumption of sphericity was not violated. The post-hoc test showed that *Hands-Down* had a significantly higher error-rate compared to all other conditions: *Baseline* ($p = 0.035$), *In-Front* ($p = 0.018$), and *Sitting* ($p = 0.001$). Also here all other conditions did not differ between each other in terms of error-rate. Figure 5 provides a more detailed overview of the results.

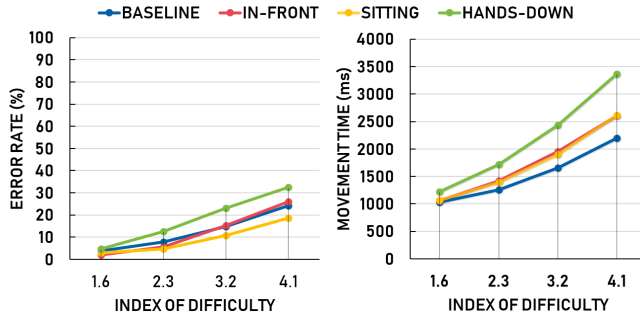


Figure 5: Error rate and movement time of each condition for each ID.

In terms of movement time (MT), the *Baseline* condition had an average MT of 1508 ms ($SD = 447$), *In-Front* 1735ms ($SD = 658$), *Sitting* with 1712 ms ($SD = 602$), and *Hands-Down* with 2148 ms ($SD = 933$). A one-way ANOVA showed a significant difference for MT between the conditions ($F_{3,33} = 17.5931, p < 0.001$). The post-hoc test found that *Hands-Down* had a significantly higher MT compared to all other conditions: *Baseline* ($p = 0.002$), *In-Front* ($p = 0.001$) and *Sitting* ($p = 0.002$). All the other pairs were not significantly different.

Figure 6 shows the accuracy measures of the Fitts' law task [34]. The results indicate a high similarity between the *Baseline*, *In-Front* and *Sitting* condition. We also found that the shortcomings

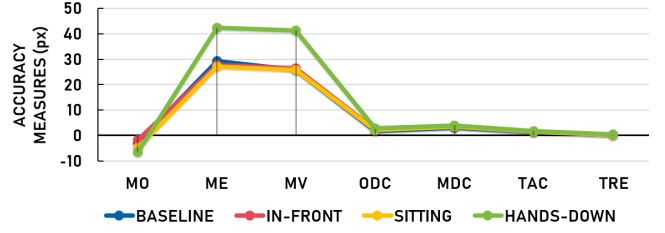


Figure 6: Fitts' law task accuracy measures for each input device. Movement offset (MO), error (ME), variability (MV), orthogonal direction change (ODC), movement direction change (MDC), task axis crossing (TAC), target re-entry (TRE)

of *Hands-Down* was mostly due to the movement error (ME) and movement variability (MV). This indicates that when participants used the *Hands-Down* condition, their cursor movement towards the target was much further away from the ideal straight line (ME) and that their motion was also not as smooth (MV), as in all other conditions. Finally, we can also report a weak correlation ($R^2 < 0.3$) between movement times and IDs for all of our conditions.

We can see that our Pocket6 conditions, *In-Front* and *Sitting*, allowed users an equal 2D pointing and clicking performance, in speed and accuracy, as the externally tracked controller *Baseline*. The *Hands-Down* condition on the other hand, performed slower and more inaccurate compared to all other conditions.

4.6 Experiment 3: 3D Manipulation

In the third experiment, we evaluated the 3D interaction by using a 3D placement task (translation only), based on the ideas of Vuibert et al. [60]. The goal of this experiment, was to see if all four conditions provide the same accuracy and speed, while participants perform continuous stop-and-move 3D motions in mid-air.

Similarly to [4], participants faced two squares displayed on the screen. These squares had to be aligned by a 3D drag-and-drop gesture. For each trial, participants moved their cursor (x - and y -axis) and grabbed a white draggable square with a tap-and-hold gesture. Once grabbed, they needed to align it in position and size with a red target square.

Participants could re-size the dragged square by moving their smartphone on the z -axis of the control space. Moving the smartphone towards the positive direction on the z -axis reduced the size of the draggable square, and the other way around. For x - and y -axis cursor movement, we used the default control rate (120 px on-screen is 1 cm in control space), for square resizing this would be too much, therefore we used smaller mapping, where a 15 px increase or decrease of the square's width corresponded to a 1 cm movement on the z -axis of the control space. If the two squares were correctly aligned, they both turned green. For correction checking, we used a tolerance of 10 px for both the position and scale. In the case of a negative match, participants had to re-grab and re-do the alignment. Once the alignment was successful, both squares disappeared and participants needed to move their 3D cursor back to the middle of the interaction area (x -, y - and z -axis), with the hint of a small cursor widget. Afterwards, the next trial was shown.

The position (x and y) of the target square was defined by an amplitude (250 or 400 px), describing a radius from the screen center and a randomly defined angle (ranging from 0 to 360°). To represent the z -axis distance, we used four target sizes (50, 95, 155, 200 px). The initial size of the draggable square was 125 px, which was also the z -axis starting position at the beginning of each trial. Participants completed a practice block first, followed by a study block. Each block contained all combinations of amplitudes and target sizes. In total, each participant completed 64 trials (4 conditions \times 2 amplitudes \times 4 target sizes \times 2 repetition).

4.6.1 Results. We removed 2.1% outliers caused by technological errors and semantic errors, like participants trying to grab the target square instead of the draggable square. The placement time for the *Baseline* condition was on average 2201 ms ($SD = 723$), for the *In-Front* condition 2334 ms ($SD = 913$), for *Sitting* 2212 ms ($SD = 861$), and for the *Hands-Down* condition with 2543 ms ($SD = 935$). A one-way ANOVA ($\alpha = .05$) did not show a significant difference for placement time between the conditions ($F_{3,33} = 5.620, p < 0.063$). The error rate was negligible for all conditions, less than 1%, since participants did not let go of the draggable square until they got the indication that the alignment was correct.

As shown, the Pocket6 conditions *In-Front*, *Sitting* and *Hands-Down*, performed comparable to the externally tracked *Baseline* condition.

4.7 User Feedback

After participants experienced all study tasks, we asked them to provide their unconstrained subjective feedback based on the overall experience across all three experiments, by rating each condition for *ease of use*, *fatigue*, *speed*, *precision*, and *overall impression* on a 7-point Likert scale (higher is better). Finally, we asked participants for additional comments, suggestions and recommendations. The goal was to learn from initial user reactions and comments with a special focus on exposing differences between our conditions

4.7.1 Results. A Friedman test indicated significant results for the ease of use ($\chi^2(3) = 20.00, p < 0.001$), fatigue ($\chi^2(3) = 18.00, p < 0.001$), speed ($\chi^2(3) = 19.00, p < 0.001$), precision ($\chi^2(3) = 19.00, p < 0.001$) and general impression ($\chi^2(3) = 18.02, p < 0.001$) depending on which condition was used, cf. Figure 7.

In depth analysis by performing a Wilcoxon Signed Rank test showed no significant difference between the *Baseline*, *In-Front* and *Sitting* conditions in any of the categories. However, there was a significant difference between the *Hands-down* and all other conditions in all categories (except for *fatigue* for the pair *In-Front* vs. *Hands-down*), see Table 1.

Participants reported that Pocket6 worked surprisingly well. Almost all participants preferred the *Baseline* or *In-Front* condition as both conditions were fast, accurate, and easy to use. Participants reported that the ergonomics of a controller seems to be very important. Additionally, they complained about the fact that the iPhone X as a bit heavier than the VR controller. Half of the participants agreed that the *Sitting* condition was very comfortable due to the fact that participants could rest their elbows. However, they also noted that this is not always required and that it could also be a

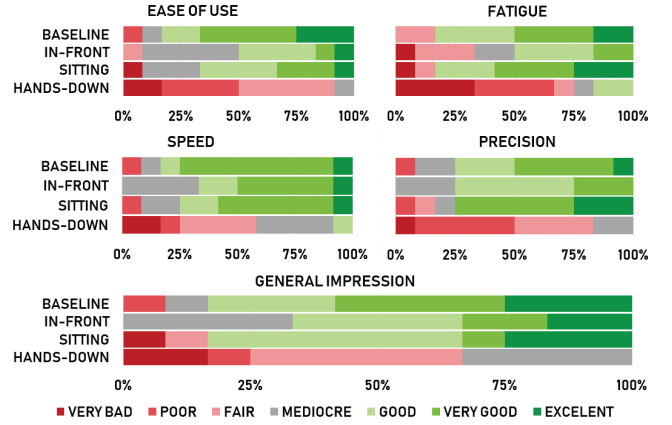


Figure 7: Subjective feedback ratings on ease-of-use, fatigue, speed, precision and general impression for each input device.

Category	Hands-Down vs. Baseline		Hands-Down vs. In-Front		Hands-Down vs. Sitting	
	Z	p	Z	p	Z	p
Ease-of-use	-3.082	0.005	-3.082	0.002	-2.000	0.006
Fatigue	-2.000	0.003	/	/	-2.000	0.006
Speed	-2.000	0.003	-3.093	0.002	-2.000	0.010
Precision	-3.016	0.003	-2.000	0.003	-2.000	0.004
General impression	-2.000	0.003	-3.088	0.002	-2.000	0.015

Table 1: Significant pairs from the subjective feedback rating reported by the Wilcoxon Signed Rank test.

serious limitation. Some participants found that resting the elbow made them lazy and that they didn't want to raise their arm once rested, e.g. to reach for items at the upper side of the screen, this caused minor frustrations.

Subjectively, all participants agreed that the *Hands-down* condition was the most fatiguing and hardest to use. This was mainly due to the long "lever" (kinematic chain from neck to finger tips) that had to be precisely adjusted. Moreover, most of the participants were not used to interact with their hands down next to the body. Participants felt that they could not interact in a *Hands-down* posture for a long time. They all agreed that the axes mapping of the *Hands-down* condition was easy to understand. One participant (P3) expressed that with the *Hands-down* condition she was missing the hand-eyes coordination, since she could not observe her hand.

Other participants explained that *Sitting* and *In-Front* both performed equally good, however they both have minor trade-offs. In the *In-Front* condition, participants could easier overshoot targets (e.g. Fitts' law tasks) and on the other side it provides more unrestricted motions. The *Sitting* condition, in contrast, seems to be more comfortable, but on the other hand it is more restricting, which was noted to be cumbersome. Some participants explicitly disliked the *Sitting* condition, explaining that it was too limited due to the elbow rest.

4.8 Discussion and Design Recommendations

Across all experiments, we found that neither the performance nor the subjective opinions of the participants varied significantly between the Pocket6 *In-Front* and *Sitting* conditions. Participants liked the *Hands-down* condition the least, and were significantly less accurate with it in comparison to the other conditions. This was a surprising result, as we had initially assumed that the relaxed *Hands-down* condition would have been the most comfortable.

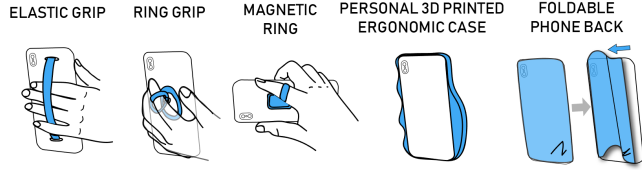


Figure 8: Proposed solutions for improving the ergonomics of a smartphone.

Participants noted that the ergonomics of larger smartphones (e.g. iPhone X) gives the impression of holding a heavier device. Although, this was only seen as a minor problem, we suggest a few simple add-ons that could be applied to address this issue. The experience of gripping the phone can be easily improved by using an off-the-shelf smartphone cover with additional handles as depicted in Figure 8.

5 APPLICATIONS

In this section and supplementary video, we demonstrate how Pocket6 can immediately be used to control a wide variety of real-world applications (i.e. Google Earth, YouTube, text editor, PowerPoint, furniture rearranging application, cf. Figure 9). It allows for 2D or 3D cursor control, through a combination of subtle mid-air gestures (e.g. spatial-translation and rotation) and touch input (e.g. taps and long-taps). This allows users to pan, drag-and-drop, and perform simultaneous point-and-zoom actions. This also allows for simple copying, pasting and selection tasks, which are often too difficult to perform using touch-only input devices. Furthermore, Pocket6 also allows for powerful object manipulation, such as the synchronous rotation and translation of 3D objects, via subtle hand motions.

Participants were given a chance to test these applications after the formal experiments, and were asked for qualitative feedback. From this, we quickly learned that participants were dissatisfied with the size of the touchpads on the commercial VR controller (1" in diameter). They felt it did not give them a fine-degree of control, and also made it difficult to perform swipe gestures. For instance, in the furniture rearranging application, this even resulted in users needing to release and re-position their thumbs to fully perform desired manipulations of 3D objects. In contrast, they found the larger smartphone touchscreen (5.8") to be much more helpful in such scenarios.

6 CONCLUSION AND FUTURE WORK

In this note, we presented an inexpensive, accessible and easy-to-use 6DoF input controller in the form of a smartphone application,

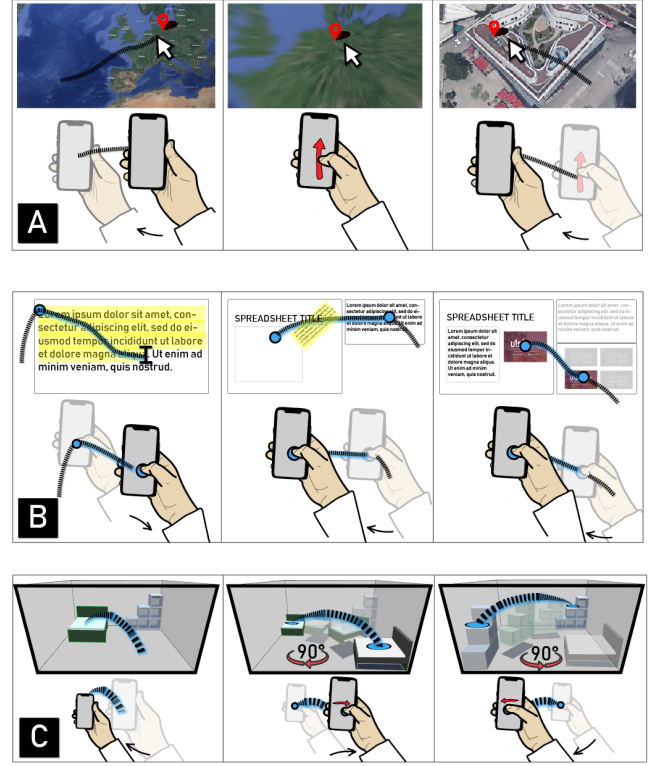


Figure 9: (a) While mid-air translation are used to control the cursor, swipe gestures allow synchronously zooming out and in of the map. (b) By clicking and translating the phone we can select text and drag it over multiple windows to the target application. (c) By translating the phone and finger swipes we can perform easy and subtle one-stroke 3D interactions, as 3D furniture rearrangement.

which uses a simple and efficient auto-calibration algorithm to adapt the users' control space whenever they change their body position, orientation, or posture. We conducted three studies to evaluate its usability in different conditions. Our results demonstrated that with Pocket6, users can achieve a high tracking accuracy in both 2D and 3D interactions. This can be done with no additional external hardware (e.g. external cameras for tracking). Furthermore, users do not need to perform large hand motions. In addition to these studies, we demonstrated how it can be used to control a variety of real-world applications.

In the future, Pocket6 could be extended in multiple ways. For example, it can be enhanced to leverage phone-enabled haptic or audio feedback as well as touch pressure-sensitivity. While we explored its use in a controlled-setting, it can also be explored in situations where users are on-the-move. Finally, we foresee that our work could allow other researchers to extend upon their very recent and interesting approaches; for instance, it could be used in collaborative 3D object manipulation [19], spatial design ideation [57] or cross-device interaction scenarios [42], which are currently limited by the use of 3DoF input-devices.

REFERENCES

- [1] Sandip Agrawal, Ionut Constandache, Shravan Gaonkar, Romit Roy Choudhury, Kevin Caves, and Frank DeRuyter. 2011. Using Mobile Phones to Write in Air. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys '11)*. ACM, New York, NY, USA, 15–28. <https://doi.org/10.1145/1999995.1999998>
- [2] Apple. 2017. ARKit Device Compatibility. Website. Retrieved April 5, 2018 from <https://developer.apple.com/library/content/documentation/DeviceInformation/Reference/iOSDeviceCompatibility/DeviceCompatibilityMatrix/DeviceCompatibilityMatrix.html>
- [3] ARToolKit. 2017. Website. Retrieved April 5, 2018 from <http://www.hitl.washington.edu/artoolkit/>
- [4] Teo Babic, Harald Reiterer, and Michael Haller. 2017. GestureDrawer: One-handed Interaction Technique for Spatial User-defined Imaginary Interfaces. In *Proceedings of the 5th Symposium on Spatial User Interaction (SUI '17)*. ACM, New York, NY, USA, 128–137. <https://doi.org/10.1145/3131277.3132185>
- [5] Rafael Ballagas, Michael Rohs, and Jennifer G. Sheridan. 2005. Sweep and Point and Shoot: Phocam-based Interactions for Large Public Displays. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems (CHI EA '05)*. ACM, New York, NY, USA, 1200–1203. <https://doi.org/10.1145/1056808.1056876>
- [6] Louis-Pierre Bergé, Emmanuel Dubois, and Mathieu Raynal. 2015. Design and Evaluation of an "Around the SmartPhone" Technique for 3D Manipulations on Distant Display. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction (SUI '15)*. ACM, New York, NY, USA, 69–78. <https://doi.org/10.1145/2788940.2788941>
- [7] Louis-Pierre Bergé, Marcos Serrano, Gary Perelman, and Emmanuel Dubois. 2014. Exploring Smartphone-based Interaction with Overview+Detail Interfaces on 3D Public Displays. In *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services (MobileHCI '14)*. ACM, New York, NY, USA, 125–134. <https://doi.org/10.1145/2628363.2628374>
- [8] Lonni Besançon, Paul Issartel, Mehdi Ammi, and Tobias Isenberg. 2017. Mouse, Tactile, and Tangible Input for 3D Manipulation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 4727–4740. <https://doi.org/10.1145/3025453.3025863>
- [9] Sebastian Boring, Dominikus Baur, Andreas Butz, Sean Gustafson, and Patrick Baudisch. 2010. Touch Projector: Mobile Interaction Through Video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 2287–2296. <https://doi.org/10.1145/1753326.1753671>
- [10] Sebastian Boring, Marko Jurmu, and Andreas Butz. 2009. Scroll, Tilt or Move It: Using Mobile Phones to Continuously Control Pointers on Large Public Displays. In *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7 (OZCHI '09)*. ACM, New York, NY, USA, 161–168. <https://doi.org/10.1145/1738826.1738853>
- [11] Andrew Bragdon, Rob DeLine, Ken Hinckley, and Meredith Ringel Morris. 2011. Code Space: Touch + Air Gesture Hybrid Interactions for Supporting Developer Meetings. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '11)*. ACM, New York, NY, USA, 212–221. <https://doi.org/10.1145/2076354.2076393>
- [12] Géry Casiez, Nicolas Roussel, and Daniel Vogel. 2012. 1 Filter: A Simple Speed-based Low-pass Filter for Noisy Input in Interactive Systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 2527–2530. <https://doi.org/10.1145/2207676.2208639>
- [13] Elie Cattani, Amélie Rochet-Capellan, Pascal Perrier, and François Bérard. 2017. Does Practice Make Perfect?. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 5619–5629. <https://doi.org/10.1145/3025453.3025585>
- [14] Raimund Dachsel and Robert Buchholz. 2009. Natural Throw and Tilt Interaction Between Mobile Phones and Distant Displays. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems (CHI EA '09)*. ACM, New York, NY, USA, 3253–3258. <https://doi.org/10.1145/1520340.1520467>
- [15] H. Debarba, L. Nedel, and A. Maciel. 2012. LOP-cursor: Fast and precise interaction with tiled displays using one hand and levels of precision. In *2012 IEEE Symposium on 3D User Interfaces (3DUI)*. 125–132. <https://doi.org/10.1109/3DUI.2012.6184196>
- [16] Henrique G. Debarba, Jerônimo G. Grandi, Anderson Maciel, Luciana Nedel, and Ronan Boulic. 2013. Disambiguation Canvas: A Precise Selection Technique for Virtual Environments. In *Human-Computer Interaction – INTERACT 2013*, Paula Kotzé, Gary Marsden, Gitte Lindgaard, Janet Wesson, and Marco Winckler (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 388–405.
- [17] Mingming Fan, Donald Patterson, and Yuanchun Shi. 2012. When Camera Meets Accelerometer: A Novel Way for 3D Interaction of Mobile Phone. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services Companion (MobileHCI '12)*. ACM, New York, NY, USA, 131–136. <https://doi.org/10.1145/2371664.2371689>
- [18] H. Graf and K. Jung. 2012. The smartphone as a 3D input device. In *2012 IEEE Second International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*. 254–257. <https://doi.org/10.1109/ICCE-Berlin.2012.6336487>
- [19] Jerônimo Gustavo Grandi, Henrique Galvan Debarba, Luciana Nedel, and Anderson Maciel. 2017. Design and Evaluation of a Handheld-based 3D User Interface for Collaborative Object Manipulation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 5881–5891. <https://doi.org/10.1145/3025453.3025935>
- [20] Martin Hachet, Joachim Pouderoux, and Pascal Guittou. 2005. A Camera-based Interface for Interaction with Mobile Handheld Computers. In *Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games (I3D '05)*. ACM, New York, NY, USA, 65–72. <https://doi.org/10.1145/1053427.1053438>
- [21] Jari Hannuksela, Pekka Sangi, and Janne Heikkilä. 2007. Vision-based Motion Estimation for Interaction with Mobile Devices. *Comput. Vis. Image Underst.* 108, 1–2 (Oct. 2007), 188–195. <https://doi.org/10.1016/j.cviu.2006.10.014>
- [22] Anders Henrysson and Mark Billinghurst. 2007. Using a Mobile Phone for 6 DOF Mesh Editing. In *Proceedings of the 8th ACM SIGCHI New Zealand Chapter's International Conference on Computer-Human Interaction: Design Centered HCI (CHINZ '07)*. ACM, New York, NY, USA, 9–16. <https://doi.org/10.1145/1278960.1278962>
- [23] Anders Henrysson, Mark Billinghurst, and Mark Ollila. 2005. Virtual Object Manipulation Using a Mobile Phone. In *Proceedings of the 2005 International Conference on Augmented Tele-existence (ICAT '05)*. ACM, New York, NY, USA, 164–171. <https://doi.org/10.1145/1152399.1152430>
- [24] Anders Henrysson and Mark Ollila. 2004. UMAR: Ubiquitous Mobile Augmented Reality. In *Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia (MUM '04)*. ACM, New York, NY, USA, 41–45. <https://doi.org/10.1145/1052380.1052387>
- [25] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed Endurance: A Metric to Quantify Arm Fatigue of Mid-air Interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1063–1072. <https://doi.org/10.1145/2556288.2557130>
- [26] ISO, 9421–9 2000. *Ergonomic requirements for office work with visual display terminals (VDTs) - Part 9: Requirements for non-keyboard input devices*. Standard. International Organization for Standardization, Geneva, CH.
- [27] Sujin Jang, Wolfgang Stuerzlinger, Satyajit Ambike, and Karthik Ramani. 2017. Modeling Cumulative Arm Fatigue in Mid-Air Interaction Based on Perceived Exertion and Kinetics of Arm Motion. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 3328–3339. <https://doi.org/10.1145/3025453.3025523>
- [28] N. Katzakis and M. Hori. 2009. Mobile Phones as 3-DOF Controllers: A Comparative Study. In *2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing*. 345–349. <https://doi.org/10.1109/DASC.2009.76>
- [29] Nicholas Katzakis, Robert J. Teather, Kiyoshi Kiyokawa, and Haruo Takemura. 2015. INSPECT: extending plane-casting for 6-DOF control. *Human-centric Computing and Information Sciences* 5, 1 (30 Jul 2015), 22. <https://doi.org/10.1186/s13673-015-0037-y>
- [30] Juha Kela, Panu Korpipää, Jani Mantjarvi, Sanna Kallio, Giuseppe Savino, Luca Jozzo, and Di Marca. 2006. Accelerometer-based Gesture Control for a Design Environment. *Personal Ubiquitous Comput.* 10, 5 (July 2006), 285–299. <https://doi.org/10.1007/s00779-005-0033-8>
- [31] David Ledo, Saul Greenberg, Nicolai Marquardt, and Sebastian Boring. 2015. Proxemic-Aware Controls: Designing Remote Controls for Ubiquitous Computing Ecologies. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '15)*. ACM, New York, NY, USA, 187–198. <https://doi.org/10.1145/2785830.2785871>
- [32] Hai-Ning Liang, Yuwei Shi, Feiyu Lu, Jizhou Yang, and Konstantinos Papangelis. 2016. VRMController: An Input Device for Navigation Activities in Virtual Reality Environments. In *Proceedings of the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry - Volume 1 (VRCAI '16)*. ACM, New York, NY, USA, 455–460. <https://doi.org/10.1145/3013971.3014005>
- [33] Mingyu Liu, Mathieu Nancel, and Daniel Vogel. 2015. Gunslinger: Subtle Arms-down Mid-air Interaction. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 63–71. <https://doi.org/10.1145/2807442.2807489>
- [34] I. Scott MacKenzie, Tatu Kauppinen, and Miika Silfverberg. 2001. Accuracy Measures for Evaluating Computer Pointing Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '01)*. ACM, New York, NY, USA, 9–16. <https://doi.org/10.1145/365024.365028>
- [35] Scott MacKenzie. 2018. Fitts' Law Software. <http://www.yorku.ca/mack/FittsLawSoftware/>
- [36] Nicolai Marquardt, Robert Diaz-Marino, Sebastian Boring, and Saul Greenberg. 2011. The Proximity Toolkit: Prototyping Proxemic Interactions in Ubiquitous Computing Ecologies. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 315–326. <https://doi.org/10.1145/2047196.2047238>
- [37] Asier Marzo, Benoît Bossavit, and Martin Hachet. 2014. Combining Multi-touch Input and Device Movement for 3D Manipulations in Mobile Augmented Reality Environments. In *Proceedings of the 2nd ACM Symposium on Spatial User Interaction (SUI '14)*. ACM, New York, NY, USA, 13–16. <https://doi.org/10.1145/2659766>

- 2659775
- [38] Mathias Mohring, Christian Lessig, and Oliver Bimber. 2004. Video See-Through AR on Consumer Cell-Phones. In *Proceedings of the 3rd IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR '04)*. IEEE Computer Society, Washington, DC, USA, 252–253. <https://doi.org/10.1109/ISMAR.2004.63>
 - [39] 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 (VRIC '13)*. ACM, New York, NY, USA, Article 12, 10 pages. <https://doi.org/10.1145/2466816.2466829>
 - [40] Mathieu Nancel, Julie Wagner, Emmanuel Pietriga, Olivier Chapuis, and Wendy Mackay. 2011. Mid-air Pan-and-zoom on Wall-sized Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 177–186. <https://doi.org/10.1145/1978942.1978969>
 - [41] OptiTrack. 2017. Motion capturing hardware. Website. Retrieved April 5, 2018 from <http://optitrack.com/hardware/>.
 - [42] Jeni Paay, Dimitrios Raptis, Jesper Kjeldskov, Mikael B. Skov, Eric V. Ruder, and Bjarke M. Lauridsen. 2017. Investigating Cross-Device Interaction Between a Handheld Device and a Large Display. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 6608–6619. <https://doi.org/10.1145/3025453.3025724>
 - [43] Michel Pahud, Ken Hinckley, Shamsi Iqbal, Abigail Sellen, and Bill Buxton. 2013. Toward Compound Navigation Tasks on Mobiles via Spatial Manipulation. In *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '13)*. ACM, New York, NY, USA, 113–122. <https://doi.org/10.1145/2493190.2493210>
 - [44] Krzysztof Pietroszek, Anastasia Kuzminykh, James R. Wallace, and Edward Lank. 2014. Smartcasting: A Discount 3D Interaction Technique for Public Displays. In *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: The Future of Design (OzCHI '14)*. ACM, New York, NY, USA, 119–128. <https://doi.org/10.1145/2686612.2686629>
 - [45] Krzysztof Pietroszek, James R. Wallace, and Edward Lank. 2015. Tiltcasting: 3D Interaction on Large Displays Using a Mobile Device. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 57–62. <https://doi.org/10.1145/2807442.2807471>
 - [46] J. Rekimoto. 1998. Matrix: a realtime object identification and registration method for augmented reality. In *Proceedings. 3rd Asia Pacific Computer Human Interaction (Cat. No.98EX110)*. 63–68. <https://doi.org/10.1109/APCHI.1998.704151>
 - [47] Michael Rohs and Antti Oulasvirta. 2008. Target Acquisition with Camera Phones when Used As Magic Lenses. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*. ACM, New York, NY, USA, 1409–1418. <https://doi.org/10.1145/1357054.1357275>
 - [48] MacKenzie I. S. 2018. *The Wiley Handbook of Human Computer Interaction* (eds K. L. Norman and J. Kirakowski). John Wiley & Sons, Ltd, Chichester, UK, Chapter Fitts' Law. <https://doi.org/10.1002/9781118976005.ch17>
 - [49] R. F. Salas-Moreno, B. Glocks, P. H. J. Kelly, and A. J. Davison. 2014. Dense planar SLAM. In *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 157–164. <https://doi.org/10.1109/ISMAR.2014.6948422>
 - [50] Julian Seifert, Andreas Bayer, and Enrico Rukzio. 2013. PointerPhone: Using Mobile Phones for Direct Pointing Interactions with Remote Displays. In *Human-Computer Interaction – INTERACT 2013*, Paula Kotzé, Gary Marsden, Gitte Lindgaard, Janet Wesson, and Marco Winckler (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 18–35.
 - [51] Bruno Siciliano. 2008. *Springer handbook of robotics*. Springer, Berlin.
 - [52] Rajinder Sodhi, Hrvoje Benko, and Andrew Wilson. 2012. LightGuide: Projected Visualizations for Hand Movement Guidance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 179–188. <https://doi.org/10.1145/2207676.2207702>
 - [53] Peng Song, Wooi Boon Goh, Chi-Wing Fu, Qiang Meng, and Pheng-Ann Heng. 2011. WYSIWYF: Exploring and Annotating Volume Data with a Tangible Handheld Device. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 1333–1342. <https://doi.org/10.1145/1978942.1979140>
 - [54] Martin Spindler, Martin Schuessler, Marcel Martsch, and Raimund Dachselt. 2014. Move Your Phone: Spatial Input-based Document Zoom & Pan on Mobile Displays Revisited. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems (CHI EA '14)*. ACM, New York, NY, USA, 515–518. <https://doi.org/10.1145/2559206.2574777>
 - [55] Martin Spindler, Martin Schuessler, Marcel Martsch, and Raimund Dachselt. 2014. Pinch-drag-flick vs. Spatial Input: Rethinking Zoom & Pan on Mobile Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1113–1122. <https://doi.org/10.1145/2556288.2557028>
 - [56] Robert J. Teather and I. Scott MacKenzie. 2014. Position vs. Velocity Control for Tilt-based Interaction. In *Proceedings of Graphics Interface 2014 (GI '14)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 51–58. <http://dl.acm.org/citation.cfm?id=2619648.2619658>
 - [57] Vinayak, Devarajan Ramanujan, Cecil Piya, and Karthik Ramani. 2016. Mo-biSweep: Exploring Spatial Design Ideation Using a Smartphone As a Hand-held Reference Plane. In *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '16)*. ACM, New York, NY, USA, 12–20. <https://doi.org/10.1145/2839462.2839490>
 - [58] Thomas Vincent, Laurence Nigay, and Takeshi Kurata. 2013. Handheld Augmented Reality: Effect of Registration Jitter on Cursor-based Pointing Techniques. In *Proceedings of the 25th Conference on L'Interaction Homme-Machine (IHM '13)*. ACM, New York, NY, USA, Article 1, 6 pages. <https://doi.org/10.1145/2534903.2534905>
 - [59] Daniel Vogel and Ravin Balakrishnan. 2005. Distant Freehand Pointing and Clicking on Very Large, High Resolution Displays. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology (UIST '05)*. ACM, New York, NY, USA, 33–42. <https://doi.org/10.1145/1095034.1095041>
 - [60] Vanessa Vuibert, Wolfgang Stuerzlinger, and Jeremy R. Cooperstock. 2015. Evaluation of Docking Task Performance Using Mid-air Interaction Techniques. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction (SUI '15)*. ACM, New York, NY, USA, 44–52. <https://doi.org/10.1145/2788940.2788950>
 - [61] Daniel Wagner, Tobias Langlotz, and Dieter Schmalstieg. 2008. Robust and Unobtrusive Marker Tracking on Mobile Phones. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR '08)*. IEEE Computer Society, Washington, DC, USA, 121–124. <https://doi.org/10.1109/ISMAR.2008.4637337>
 - [62] Jingtao Wang, Shumin Zhai, and John Canny. 2006. Camera Phone Based Motion Sensing: Interaction Techniques, Applications and Performance Study. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology (UIST '06)*. ACM, New York, NY, USA, 101–110. <https://doi.org/10.1145/1166253.1166270>
 - [63] Jiahui Wu, Gang Pan, Daqing Zhang, Guande Qi, and Shijian Li. 2009. Gesture Recognition with a 3-D Accelerometer. In *Proceedings of the 6th International Conference on Ubiquitous Intelligence and Computing (UIC '09)*. Springer-Verlag, Berlin, Heidelberg, 25–38. https://doi.org/10.1007/978-3-642-02830-4_4