FaceTouch: Enabling Touch Interaction in Display Fixed UIs for Mobile Virtual Reality

Jan Gugenheimer, David Dobbelstein, Christian Winkler; Gabriel Haas, Enrico Rukzio
Institute of Media Informatics, Ulm University, Ulm, Germany
<firstname>.<lastname>@uni-ulm.de

Figure 1. (a) A user interacting with FaceTouch, a multi-touch surface mounted on the back of a VR HMD. FaceTouch allows for precise interactions which can be used to implement applications such as text entry (b) or 3D modeling (c). Leveraging the sense of proprioception a user is able to blindly interact with control elements such as used in a gamepad to control a shooter game (d).

ABSTRACT
We present FaceTouch, a novel interaction concept for mobile Virtual Reality (VR) head-mounted displays (HMDs) that leverages the backside as a touch-sensitive surface. With FaceTouch, the user can point at and select virtual content inside their field-of-view by touching the corresponding location at the backside of the HMD utilizing their sense of proprioception. This allows for rich interaction (e.g. gestures) in mobile and nomadic scenarios without having to carry additional accessories (e.g. a gamepad). We built a prototype of FaceTouch and conducted two user studies. In the first study we measured the precision of FaceTouch in a display-fixed target selection task using three different selection techniques showing a low error rate of ≈2% indicate the viability for everyday usage. To assess the impact of different mounting positions on the user performance we conducted a second study. We compared three mounting positions of the touchpad (face, hand and side) showing that mounting the touchpad at the back of the HMD resulted in a significantly lower error rate, lower selection time and higher usability. Finally, we present interaction techniques and three example applications that explore the FaceTouch design space.

Author Keywords
Back-of-device interaction; Mobile VR; VR interaction; Virtual Reality; Nomadic VR; VR input

INTRODUCTION
Virtual Reality (VR) head-mounted displays (HMD) are having a consumer revival with several major companies such as Facebook, Sony and Samsung releasing their consumer devices this year. In contrast to VR HMDs that are operated by a computer (such as OculusRift and HTC Vive), mobile HMDs have been presented which are operated solely by a mobile phone (e.g. Samsung GearVR and Google Cardboard). These mobile VR HMDs allow new usage scenarios where users can access Immersive Virtual Environments (IVEs) anywhere they want. Based on aspects of nomadic computing [17], we define this as nomadic VR.

Due to the omnipresence of mobile phones and the relatively low price, mobile VR HMDs (e.g. Google CardBoard) are expected to penetrate the consumer market more easily. However, current VR input research such as [1] and consumer products are focusing on stationary HMDs and input modalities that would not be available in nomadic scenarios. These include the instrumentation of the environment (e.g. Oculus’ positional tracking, HTC VIVE’s Lighthouse) or the usage of peripheral devices like 3D mice or game controllers. Hand tracking technology such as the Leap Motion strives for enabling “natural” interaction inside an IVE and lead to a higher level of immersion for certain scenarios (e.g. immersive experiences) but discounts utilitarian interactions such as browsing a menu or entering text, where the goal is on performance and less on immersion. We argue that interaction for VR should not only focus on enabling those “natural” interaction concepts but also enable a “super natural” interaction where users can interact and manipulate the virtual environment with little physical effort and enable interactions beyond human capability.

*now at Daimler Protics GmbH
We therefore investigate the concept of touch interaction inside an IVE as a first step towards that direction.

Current mobile VR UIs are designed to be operated using Head-Rotation with a crosshair cursor or a gamepad. Since gamepads are not bundled with any mobile HMD (and do not fit the nomadic usage) the most targeted and used selection technique is Head-Rotation. This leads to a limitation in the UI design space. With HeadRotation, a crosshair cursor is centered in the middle of the view, so that the user can aim at the target by rotating their head and select by using another means of input, such as a button or touch panel at the side of the VR device. The area of view has to be centered around the target location and as an implication, it is not possible to design display-fixed user interface elements (e.g., targets that are always at the bottom of the display). For this reason, current UI elements are implemented to be at a fixed location in 3D space (world-fixed UI). This forces either the content creator to embed every possible UI element (consider a keyboard for text input) inside the 3D scene or the user to leave their current scene to control UI elements (e.g., Samsung GearVR settings menu).

FaceTouch

To address these shortcomings, we present FaceTouch, an interaction technique for mobile VR HMDs leveraging the backside of the HMD as a touch surface (see Fig. 1). Adding touch input capabilities to the backside allows for direct interaction with virtual content inside the users field-of-view by selecting the corresponding point on the touch surface. Users cannot see their hands while wearing the HMD, but due to their proprioceptive senses [20] they have a good estimate of their limbs in relation to their body. Supported by visual feedback as soon as fingers are touching the surface, as well as their kinesthetic memory, users find in FaceTouch a fast and precise alternative interaction technique for nomadic VR scenarios that does not require them to carry an additional accessory (e.g., a gamepad).

In order to explore the design space we built a hardware prototype consisting of an Oculus Rift and a 7 inch capacitive touchpad mounted to the backside (see Fig. 3). We ran two user studies to investigate the precision and interaction time of FaceTouch for display-fixed UIs and measure the impact of the mounting position on those factors. In a first user study (n=18) we conducted a target selection task in a display-fixed condition showing a possible throughput [22] of \( \approx 2.16 \text{ bits/s} \). Furthermore, we present a selection point cloud, showing how precise users can point at targets relying only on proprioception. In a second user study (n=18), we investigated the impact of the mounting position on performance, comparing three different locations (face, hand and side) and showing a significantly lower error rate and lower selection time when mounting the touchpad on the backside of the HMD, justifying our design decision for FaceTouch.

**Contributions**

The main contributions of this paper are:

- The concept of FaceTouch, an interaction technique for mobile VR HMDs allowing for fast and precise interaction in nomadic VR scenarios. It can be used on its own or combined with HeadRotation to further enrich the input space in mobile VR.
- Showing the feasibility of FaceTouch for display-fixed user interfaces, offering a low selection error rate (\( \approx 3\% \)) and fast selection time (\( \approx 1.49 \text{ s} \)), making it viable for everyday usage.
- Comparing three different mounting positions of the touchpad and showing the advantages (\( \approx 8\% \) less errors then hand and \( \approx 29\% \) less then side) and user preference for the face mounting location.
- Exploration of the design space of FaceTouch through the implementation of three example applications (gaming controls, text input, and 3D content manipulation) showing how the interaction can be utilized in display-fixed as well as world-fixed VR applications.

**Related Work**

Our work is related to the research fields of back-of-device interaction, proprioceptive interaction and input techniques for IVEs.

**Back-of-Device Interaction**

In order to eliminate finger occlusion during touch interaction, researchers proposed back-of-device interaction [14, 18, 35, 2] which leverages the backside of a mobile device as an input surface.

Several implementations and prototypes where proposed which either used physical buttons on the backside [14, 18] or used the backside as a touch surface [31, 35]. Wigdor et al. enhanced the concept by introducing "pseudo-transparency" which allowed the users to see a representation of their hand and fingers allowing the users to precisely interact with the content independent of finger sizes [37]. Furthermore, Baudisch et al. showed that the concept of back-of-device interaction works independent of device sizes [2]. Wigdor et al., applied the concept further to stationary devices such as a tabletop [38]. Without seeing their hands and using only the sense of proprioception, participants interacted with a tabletop display by selecting targets under the table.

FaceTouch extends the field by being the first work utilizing back-of-device interaction in VR. In contrast to existing techniques, the user is completely visually decoupled from their body and by that means not able to see their arms while approaching a target. This forces the user to rely even more on proprioception to interact with the content.

**Proprioceptive Interaction**

The human capability of knowing the position and relation of the own body and its several body parts in space is called proprioception [3]. It usually complements the visual sense when reaching for a target, but even when being blindfolded from their physical environment, users can utilize their proprioceptive sense especially well to reach parts of their own body, such as being able to blindly touch their own nose [15].

Wolf et al. showed that due to the proprioceptive sense, participants were able to select targets on the backside of an iPad without visual feedback having no significant decrease in accuracy compared to visual feedback [39]. Serrano et al. explored the design space of "hand-to-face" input, where participants used gestures such as strokes on their cheeks for interacting with an HMD [33]. Lopes et al. showed how the sense of proprioception can be used as an output modality [20]. Similar to FaceTouch, most work in the field of back-of-device interaction leverages the sense of proprioception. A novelty of FaceTouch is that a back-of-device touchpad is attached to the user’s body and as a result the user can utilize proprioception while being immersed in a virtual environment. Also the user’s hands are not constrained by holding a device and can unrestrictedly be used for touch interaction.
Further, the use of proprioception was often explored in IVEs [24, 7, 19]. Mine at al. showed the benefits of proprioception in IVEs by letting participants interact with physical props in the non-dominant hand [24]. Similar to this approach, Lindeman et al. used a paddle in the non-dominant hand to leverage proprioception and passive haptic feedback in virtual hand metaphors [19].

**Input Techniques for Virtual Environments**

Besides novel feedback mechanisms [9, 10], a big part of recent VR research revolves around interaction concepts. The focus of interaction concepts for IVEs in related work is mostly on 3D interaction techniques [1] which can be classified as *exocentric* and *egocentric* interaction metaphors [28], distinguishing between whether the user interacts in a first-person view (*egocentric*) or a third-person view (*exocentric*) with the environment. Our focus will be on *egocentric* interaction concepts of which the most prevalent are the virtual hand and virtual pointer metaphor [1, 29].

The virtual hand metaphor is applied by tracking the user’s hand and creating a visual representation of it allowing the user to interact with content within arm’s reach [21]. Lindeman et al. presented how using a physical paddle in the user’s non-dominant hand to create passive haptic feedback can increase user performance for hand metaphor selection tasks [19]. *FaceTouch* offers the same advantages in terms of passive haptic feedback without forcing the user to hold a physical proxy. To enable virtual hand metaphor interaction with UI elements, researchers proposed concepts such as GoGo [27] or HOMER [4] which apply non-linear scaling of the hand position.

Virtual pointer metaphors rely on casting a ray into the virtual scene to enable user interaction [23]. Several techniques were proposed to determine the ray’s orientation which mostly rely on tracking the user’s hand similar to the virtual hand metaphor. The orientation of the ray can either be controlled by the hand position and wrist orientation or as a ray cast from the user’s viewpoint through the hand [26]. Different approaches combine either both hands [24] or use eye tracking [36]. The *HeadRotation* interaction of Samsung’s GearVR can be considered a virtual pointer metaphor where the ray is cast perpendicular to the center of the user’s viewpoint.

In contrast to previous work, *FaceTouch* enables direct interaction with content in and outside of the user’s vicinity without external tracking or additional accessories (as had been used in [30, 25]) and can be easily implemented in future mobile VR devices. Furthermore, *FaceTouch* offers passive haptic feedback which typically results in a higher selection performance [6].

**INTERACTION CONCEPT**

The basic principle of *FaceTouch* is to leverage the large unexploited space on the backside of current HMDs as a touch sensitive surface. This allows for the creation of a mapping between the physical touch surface in front of the user and their field-of-view within the IVE. By touching the surface, the user is touching a virtual plane within their field-of-view (see Fig. 2) with the same ratio and resolution as the physical touchpad resulting in a 1:1 direct mapping of physical touch and virtual selection. When aiming for a target, users can see the touch position of their fingers visualized on the virtual plane as soon as touching the surface. We refer to this step as *LandOn*. To commit a selection, we use two different techniques that can both complement each other for different selections. With *LiftOff*, a selection is committed when lifting a finger above a target, while with *PressOn*, a target is selected by applying pressure. Both techniques allow the user to correct the position of a finger on the virtual plane, before committing the selection. User interface elements for *FaceTouch* can be both: fixed to the display or to the world [8] (see Fig. 2).

**World-fixed UIs**

In current mobile VR HMDs, such as Samsung Gear VR, user interface elements are fixed within the virtual world and selectable by rotating the head and thereby turning the target into the center of the user’s view. This concept of interaction is suitable for UIs which try to immerse the user into the scene. However, it also poses the drawback that only elements within the centered focus (e.g. a crosshair in the center of the display) can be selected and a lot of head rotation is required for successive selections. With *FaceTouch*, *world-fixed* user interface elements can be selected alike, however the user does not have to center their view at the target. It is possible to select targets anywhere within the field-of-view by selecting the corresponding point on the virtual plane. Hence, users can keep their focus wherever they like.

**Display-fixed UIs**

In addition to *world-fixed* interfaces, *FaceTouch* allows to place *display-fixed* UI elements. These are always attached to the virtual plane and are independent of the user’s orientation (being always inside the users field-of-view). Examples for this are menu buttons that prove to be useful throughout interaction, such as reverting the last action in a modeling software, opening a settings menu, or virtual controls for gaming applications (more details in the Applications section). *Display-fixed UI elements can be transparent to not occlude the field-of-view or even completely hidden for more experienced users. These kind of interfaces are crucial to realize utilitarian concepts such as data selection or text entry which focus more on user performance than on immersion. Therefore, the rest of this paper will focus on investigating parameters and performances with *display-fixed UIs*.

**IMPLEMENTATION**

We built a hardware prototype of *FaceTouch* by mounting a 7 inch capacitive touchpad (15.5cm x 9.8cm) to the backside of a Oculus Rift DK2 (see Fig. 3). Even though we do not consider the Oculus Rift a mobile VR HMD since it has to be connected to a computer, it allowed us to easily integrate the rest of the hardware and was sufficient for our study designs. The touchpad is embedded in a 3D-printed case and attached to the HMD via 5 small buttons to enable the detection of finger presses on the touchpad. An Arduino Pro Mini is used to control these buttons. The *side*
touchpad was mounted on the right side of the HMD to simulate an often used mounting location for HMDs which is considered ergonomic (e.g. GearVR and Google Glass). The side touchpad has the same resolution and aspect ratio as the face touchpad. The size is approximately 10.8cm x 6.8cm. Both touchpad were picked so that they would offer as much touch space as possible for the mounting position used. Oculus Rift, the touchpad and the Arduino are tethered to a computer running Windows 8.1. The VR environments are rendered with Unity 5.0.1.

**DISPLAY-FIXED UI - USER-STUDY**

To show that FaceTouch can be used on daily basis with mobile/nomadic VR HMDs we ran a user study which simulates the interaction with display-fixed interfaces. We conducted a target selection user study for display-fixed UIs to investigate parameters relevant for FaceTouch. Since users rely on proprioception, we were interested in how accurate and fast users could hit targets of different sizes and locations, especially without visual feedback. Depending on size and distance, we expect users to get close to the target while blindly attempting a selection, but not being able to accurately select the target. For this reason we compared LandOn, as a selection technique without visual feedback as a baseline to LiftOff and PressOn. The latter two allow for the correction of the initial selection by first visualizing the touch location and requiring an additional commit method afterwards.

By positioning the virtual touch plane at the actual distance of the physical surface, we expect less interference with the proprioceptive sense. However, the Oculus guidelines [40] suggest display-fixed virtual planes to fill out only a third of the field of view leading to less “eye strain”. For that reason, we were also interested in the effect of changing the virtual plane distance.

**Study Design**

The study was conducted as a target selection task using a repeated measures factorial design with three independent variables. As independent variables we chose commit method (LandOn, LiftOff and PressOn), plane distance (NearPlane, MidPlane and FarPlane) and target size (small and large).

**Commit method.** We implemented three methods to commit a selection. With LandOn, a target is immediately selected at the initial point of contact of a finger. By this, no visual feedback is given prior to selection. LiftOff, selects the target that was touched when lifting the finger from the surface, while PressOn selects the target below the finger when physical pressure is applied to the touchpad. For LiftOff and PressOn, a cursor is presented on the virtual plane as visual feedback to represent the finger.

**Plane distance.** We used three different ratios for the field-of-view and the size of the virtual plane. NearPlane positioned the virtual plane at the same virtual distance as the touchpad was attached to the HMD. FarPlane positioned the virtual plane at a distance to fill out approximately a third of the field of view, as suggested by the guidelines of OculusVR [40]. The MidPlane was positioned in-between NearPlane and FarPlane, filling out approximately half of the field-of-view.

**Target size.** The small circular targets were picked based on the Android Design Guidelines for the smallest target having the size of 48dp (density-independent pixels) approximately 7.8mm. large targets received double the size (96dp approximately 15.6mm).

This resulted in nine combinations (3 commit methods x 3 plane distances) which were presented to the participants using a 9x9 Latin square for counterbalancing. Target size was randomized together with the target position as described in the Procedure.

The dependent variables were selection time, error rate and simulator sickness. The latter was measured using the RSSQ (Revised Simulator Sickness Questionnaire) [16]. We included the simulator sickness since we were particularly interested in the subscale “Ocular Discomfort” and expected the plane distance to influence this.

**Procedure**

For the first user study we only used the face mounting position. All participants performed a target selection task whilst wearing the FaceTouch prototype and sitting on a chair. Participants were instructed to lean back on the chair and were not allowed to rest their arms on a table to simulate the nomadic scenario. To begin with, participants were introduced to the concept of FaceTouch and filled out a demographic questionnaire. Based on the Latin square, each combination (commit method and plane distance) was presented and explained to the participants. Each participant filled out the RSSQ for simulator sickness before and after completing the target selection task with each combination. Participants were allowed to practice with each combination until they felt comfortable. At the end each participant filled out a final questionnaire comparing the presented combinations.

The target selection task consisted of 12 circular targets arranged in a 4x3 cellular grid across the virtual plane (Fig. 4). Similar
to Lubos et al. [21], participants started with selecting the start button before each target which was located in the center of the plane having the target size small. This started the timer and randomly spawned a target in the center of one of the 12 cells. This allowed us not having to use a perfect circular arrangement of targets but cover the full surface of the touchpad (also the corners) and still have a fair measurement of time. Each cell was repeated 3 times with both target sizes resulting in at least six targets per cell and at least 72 targets per combination. If a participant failed to successfully select a target the target was repeated at a later point in time (similar to [2] this repetition was not applied for LandOn since a high error rate made it impracticable). For each participant, the study took on average 1.5 hours.

**Participants**
We randomly recruited 18 participants (12 male, 6 female) from our institution with an average age of 27 (range: 21 to 33). All had an academic background being either students or had studied at the university. On average participants had been using touchscreens for 10 years (range: 3 to 12). Eight of the participants had never used an HMD before. Each participant received 10 currency.

**Results**
Our analysis is based on 18 participants selecting targets of 2 sizes on 12 locations with 3 different plane distances using 3 different commit methods each with 3 repetitions resulting in over 11664 selections.

**Error Rate**
An error was defined as a selection attempt which did not hit the target (selecting the start button was not taken into consideration). Figure 5 shows the average error rate for each commit method with each plane distance and each target size. A 3x3x2 (commit method x plane distance x target size) repeated measures ANOVA (Greenhouse Geisser corrected in case of violation of sphericity) showed significant main effects for commit method ($F(1.078,18.332)=634.822, p<.001, \eta^2=0.97$), plane distance ($F(2,34)=8.928, p<.001, \eta^2=0.24$) and target size ($F(1,17)=801.810, p<.001, \eta^2=0.97$). We also found significant interaction effects for target size x commit method ($F(1.141,19.402)=437.581, p<.01, \eta^2=0.96$).

As we expected, pairwise comparisons (Bonferroni corrected) revealed that participants made significantly more errors ($p<.001$) using LandOn (M=54.7%, SD=9%) than PressOn (M=1.8%, SD=1.9%) and significantly ($p<.001$) more using LandOn than LiftOff (M=2.2%, SD=1.8%). It is worth pointing out that the average LandOn error rates for the targets close to the start button (target 5 and 6 on Fig. 7) were only at 8%. This indicates that the precision drastically reduces when the user had to cover longer distances blindly.

A second interesting finding was that participants made significantly ($p<.05$) more errors using the NearPlane (M=20.9%, SD=4%) compared to the MidPlane (M=18.4%, SD=4%). One has to keep in mind that the plane distance only changed the visual target size, not the actual target size on the touchpad. This showed similar to prior work [41] that the target size which is presented to the user, significantly influences the accuracy of the pointing, even if the actual touch area stays the same. Finally, we found a significantly ($p<.001$) higher error rate of participants selecting small targets (M=25.6%, SD=3.8%) compared to large targets (M=13.6%, SD=2.9%).

**Selection Time**
As the selection time we defined the time between selecting the start button and the target. Only successful attempts were taken into consideration. Figure 6 shows the average selection time for each commit method, plane distance and target size. We excluded LandOn from the analysis since it resulted in a too high error rate. A 2x3x2 (commit method x plane distance x target size) repeated measures ANOVA (Greenhouse Geisser corrected in case of violation of sphericity) showed significant main effects for plane distance ($F(2,34)=8.928, p<.05, \eta^2=0.17$) and target size ($F(1,17)=345.773, p<.001, \eta^2=0.95$).

Confirming with Fitts’ Law, pairwise comparisons (Bonferroni corrected) revealed that participants were significantly ($p<.001$) faster in selecting large targets (M=1.22s, SD=0.17s) than small targets (M=1.51s, SD=0.19s). For comparisons, we calculated the mean selection time of LandOn (M=0.84s, SD=0.14s). Unlike for the error rate, plane distance had no significant influence on the selection time.

Using this data we calculated an average throughput (following the methodology of [34]) for LiftOff of around (M=2.16 bps, SD=0.28 bps). The average throughput values for the mouse range from 3.7bps to 4.9bps [34] whereas touch has an average of 6.95bps [32].

**LandOn Precision**
Bonferroni corrected pairwise comparisons of means revealed that within their three attempts, participants’ touches resulted in a significantly ($p<.001$) higher amount of overshoots with small targets (M=1.44, SD=0.2) than with large targets (M=1.19, SD=0.29). Additionally, participants’ touches resulted in a significantly ($p<.001$) higher amount of overshoots using NearPlane (M=1.6, SD=0.25) than MidPlane (M=1.3, SD=0.25) and significantly ($p<.001$) higher amount of overshoots using NearPlane than FarPlane (M=1.0, SD=0.4). To be able to understand and optimize the interaction using LandOn, we did an in-depth analysis of the selection locations. We were hoping to get a better insight into the level of accuracy people are able to achieve using the proprioceptive sense and how participants were using FaceTouch. We logged the location participants touched and defined an overshoot as a touch with a distance more than the length of the direct path. A 2x3x12 (target size x plane distance x target location) repeated measures ANOVA (Greenhouse Geisser corrected in case of violation of sphericity) on the number of overshoots (within the three attempts) showed a significant main effect for target size ($F(1,17)=24.179, p<.001, \eta^2=0.58$), plane distance ($F(2,34)=17.965, p<.001, \eta^2=0.51$) and target location ($F(11,187)=20.377, p<.001, \eta^2=0.54$). Furthermore, there were significant interactions between target size x target location ($F(11,187)=2.103, p<.05, \eta^2=0.11$) and plane distance x target location ($F(22,374)=3.159, p<.001, \eta^2=0.16$).

To explore the differences between the cells, we numbered each cell of the target location (see Fig. 7). Pairwise comparisons of means between each cell revealed significant differences in the amount of overshoots. We could divide the cells in two groups, an overshoot (cells 2,3,6,7,10,11) and an undershoot group (cells 1,4,5,8,9,12), each containing half of the cells. Figure 7 shows the touch locations for small targets and MidPlane where the centroids for failed and successful selections are represented as a triangle, respectively a circle. One can easily see the two groups by comparing the relation between the success and the fail
centroids to the center. In the overshoot group the fail centroids are always further away from the start location, whereby in the undershoot group the fail centroids are between the start location and the target. This overshooting is related to the distance the users finger has to travel. These findings show that when relying solely on proprioception, users tend to overestimate their movement over longer distances, resulting in an undershooting and underestimate it when the target is close.

In a next step we created a function which calculates the optimal target size so 95% of the touch points would end up to be successful (this is only a rough estimate since the target size itself can influence performance [41]). The optimal target size would have a diameter of around 370px (30.06mm) which is smaller than targets of Wigdor et al. [38]. We assume this is due to the fact that people have a better sense of proprioception in their facial area than with a stretched out arm under the table.

**Usability Data**

In a final questionnaire we let participants rank the commit method and plane distance based on their preference. Participants ranked LiftOff unanimously to be the commit method they would like to use (second was PressOn). Furthermore, participants (17 votes) voted MidPlane to be the most comfortable to use followed by NearPlane and FarPlane. Commenting on open-ended questions, participants mentioned that they thought FaceTouch was a “great idea” (P16), worked “surprisingly well” (P10) and was “fast to learn” (P7). Analyzing the simulator sickness data we did not find any occurrence of simulator sickness (M=1.09, SD=0.56 on a practical scale of −8.44 to 82.04 [16]) nor significant differences for the different variables.

**Discussion**

Our research question for the first user study was to find out if FaceTouch is usable for display-fixed UIs and how the parameters commit method, plane distance, target size interact with the performance.

**LiftOff.** The low error rate and overall short selection time shows that LiftOff is overall suitable to interact with current UIs for VR HMDs. The UI elements can be picked being even smaller than the small targets (7.8mm), since the error rate was around 2.2%. However, calculating the perfect sizes needs further investigation. The touch data for LiftOff showed that participants mostly started from the center of the touchpad (on average 460px away from the target location) and did not try to place the initial touch close to the target. So for precise interaction, participants need one reference point where they start their movement and start seeing the position on the touchpad. We leveraged this in the implementation of one of our example applications (Text Entry Fig. 13) by splitting the keyboard into two parts and allowing the user to have one reference point for each hand leading to a reduced overall movement.

**PressOn.** The overall performance in terms of error rate and selection time of PressOn was similar to LiftOff, indicating that it would also be a valid choice for interacting with mobile VR HMDs. During the tasks, most participants never lifted the finger from the touchpad preferring to have the visual cue of the current touch location similar as for LiftOff. The biggest downside of PressOn was that pressing down on the touchpad resulted in the IVE to “shaking” and led to a higher physical demand. This only occurred in the PressOn condition, all other conditions had no negative effect since we used a capacitive touchpad that needs no pressure. However, this did not lead to a higher simulator sickness but was reported as being “uncomfortable”. In a future prototype this can be solved using technology such as “ForceTouch” introduced by Apple.

As expected, LandOn performed significantly worse in terms of error rate in comparison to the other two commit methods. Nevertheless, it indicated a lower selection time (M=0.84s,
TOUCHPAD POSITIONING - USER STUDY

After showing the precision which FaceTouch offers with display-fixed UIs on the face mounting position we wanted to explore alternative mounting position of the touchpad and measure their impact on the users performance. We decided to compare three mounting positions (face, hand, side). We selected those positions since we expected face to have the highest level of perception and therefore the highest accuracy, hand because of its comfortable position over long use and side as a baseline to compare against the current state of the art of controlling HMDs with a touchpad at the temple (e.g. GearVR or Google Glass). Based on the optimal parameters for target size and target location we determined in the first user study, we conducted a target selection study with display-fixed UIs placing the touchpad either on the back of the HMD (face), in the hand of the user (hand) or similar to the GearVR on the side of the HMD (side) (see Fig. 8). The goal was to determine if placing the touchpad on the backside of the HMD would affect the the proprioceptive cues more compared to the other two positions.

Study Design

The study was conducted using a repeated measures factorial design with one independent variable (mounting position) having three levels (face, hand and side). As a selection technique we used LandOn and LiftOff however did not compare between those since we used different target sizes which were the optimal from the first user study (LandOn with large and LiftOff with small). We decided to use large for LandOn to be able to compare the results for hand and side with the first study. We omitted PressOn from the study since it yield similar results to LiftOff. The plane distance was MidPlane. The mounting position and commit method were counterbalanced.

The dependent variables were selection time, error rate, usability and workload. Usability was measured using the SUS questionnaire [5] and workload using the raw NASA-TLX [12]. The touchpad on the side had the same aspect ratio and resolution as the face but was smaller in size (10.8 cm x 6.8 cm) to fit on the side of the HMD. The mapping from the touchpad on the side to the input plane in front of the user was evaluated in an informal pre-study with several colleges from the institution and set fix for all participants (from the users perspective back being right and front being left). For the hand condition the touchpad from face was taken out and put into a case which the participant would hold in his non dominant hand an interact using the dominant hand. Other than this, the same apparatus as in the first study was used.

Procedure

The same target selection task as in the first user study for display-fixed UIs was used. Participants were able to practice as long as they wanted and started with LandOn or LiftOff (counterbalanced). Each of the 12 targets were selected three times. After both commit method with each mounting position was done participants filled out the SUS and NASA-TLX questionnaire. At the end of the study participants ranked each mounting position in terms of comfort and could comment on the positioning. The whole study took on average 45 minutes.

Participants

We randomly recruited 18 participants (14 male, 4 female) with an average age of 26 (range: 20 to 36) and all having an academic background being either students or employed at the institute. On average participants had 6 years experience using touchscreens and 7 had experience in using VR HMDs. Each participant received 10 currency.

Results

Error Rate: An error was defined similar to the first study. Figure 10 shows the distribution of the error rate for each mounting position. A one factorial repeated measures ANOVA showed a significant effect for mounting position \( F(2,34)=38.276, p<.001, \eta^2=0.69 \) using LandOn. Bonferroni corrected pairwise comparisons revealed that face \( M=0.35, SD=0.1 \) had a significant lower error rate than hand \( p<.05 \) and side \( M=0.65, SD=0.1 \) and side \( M=0.65, SD=0.1 \) had a significant lower error rate than hand \( p<.05 \).
As a further metric for the precision of the touches for LandOn and LiftOff_we calculated the euclidean distance for each touch point from its target center (see Fig. 9). This gives an estimate of how scattered points were and is a finer measure the just the boolean of hit or miss. A one factorial repeated measures ANOVA showed a significant effect for mounting position (F(2,34)=69.302, p<.001, η²=0.80). Bonferroni corrected pairwise comparisons revealed that face (M=91.70 px, SD=10.5 px) had a significant lower scatter compared to hand (M=110.81 px, SD=18.40 px, p<.001) and side (M=160.70 px, SD=28.84 px). Furthermore, hand had a significant lower scatter compared to side (p<.001). Combining these results with the significant lower error rate showed that participants could easier locate the targets when the touchpad was positioned at the face.

Selection Time: Similar to the first study, we measured the time between selecting the start button and selecting the target. Only successful attempts were taken into consideration. Figure 10 shows the average selection time for each mounting position using LandOn and LiftOff. A one factorial repeated measures ANOVA showed a significant effect for mounting position (F(2,34)=3.159, p<.01, η²=0.01) and NASA-TLX questionnaire (F(2,34)=25.134, p<.001, η²=0.60). Bonferroni corrected pairwise comparisons revealed a significant higher SUS score of face (M=79.86, SD=10.72) versus side (M=51.11, SD=19.40) (p<.001) and hand (M=76.11, SD=14.84) versus side (p<.001). Furthermore, side (M=27.11, SD=5.48) had a significant higher workload compared to face (M=17.22, SD=4.21) and hand (M=18, SD=5.92) (p<.001). Overall, face had the highest SUS rating and lowest NASA-TLX workload score. This shows that users preferred the face location in terms of usability and workload.

To measure fatigue, we let participants state their physical demand on a 7 point Likert scale (subscale of the NASA-TLX). A one factorial ANOVA revealed a significant difference between the mounting position for physical demand (F(2,34)=8.721, p<.001, η²=0.34). Bonferroni corrected pairwise comparisons revealed a significant lower physical demand of face (M=3.1, SD=1.7) versus side (M=3.8, SD=1.35) (p<.01) and hand (M=2.2, SD=1.4) versus side (p<.01).

Discussion

The goal of the positioning study was to measure the impact of the location of the touchpad for LandOn and LiftOff. The LiftOff commit method showed no big differences between the different mounting positions even though face was slightly better in terms of error rate and selection time compared to hand and side. Interacting using LiftOff benefits from the visualization and therefore does not rely on the proprioceptive sense that much.

The biggest difference for the mounting position were found in the LandOn condition. Placing the touchpad at the backside of the HMD(face) resulted in the overall best result (significant lower errors, scatter of touchpoints and highest SUS and lowest workload). Participants mentioned that they had a better “understanding” and “perception” when trying to blindly find the touch points. This probably results from the fact that the proprioceptive sense works better around the facial location and has more cues that the participants know the location of (eyes, nose, mouth etc.). Holding the touchpad in the hands (hand) users only have two known relation points, the supporting hand and an approximate of the location from the finger touching. Participants also mentioned it was more difficult to coordinate those two actions (holding still and touching) which is easier in the face position. When positioning the touchpad on the side participants had to create a mental mapping from the physical touchpad.
located perpendicular to the virtual floating pad. Participants mentioned that this was inherently difficult (we let participants experience the reversed mapping as well but none perceived it as better fitting) whereby placing the touchpad at the back of the HMD (face) allowed "almost directly touching" the targets.

**Fatigue**

One of the big concerns when designing interaction for IVES is the level of fatigue users will experience when interacting. Hand tracking technology such as the Leap Motion are a negative example here because of the 'touching the void' effect [6]. Furthermore, [11] and [13] showed that having the 'elbows tucked in' or 'bent the arm' results in significant less fatigue than stretching the arm away from the body. However, the last one is necessary for most hand tracking devices since they are attached on the backside of the HMD and the hands must be in their FoV.

Using FaceTouch, fatigue occurred after our user studies that took on average over 1h. However, the motivation for FT is that such an interaction is being often used for short utilitarian purposes. Furthermore, when comparing against the currently wide spread touchpad at the temple (side), FaceTouch resulted in significant lower physical demand. To further increase the comfort of the interaction, participants started already to apply techniques on how to support their arms or heads to avoid fatigue effects (e.g. 'The Thinker Pose', lean back into the chair wrap the non-dominant arm around your chest and rest the dominant arm on it). This position can easily be held over the envisioned period of interaction compared to stretching the arms away from the body [11, 13].

When using FaceTouch over a longer period of time participants mentioned to expand the concept and allow to detach the touchpad and be able to hold it in the hand and using it with LiftOff. This would lower the fatigue of holding the arm over a longer period and allow for a more comfortable position. However, for small and fast interactions, participants (8) preferred using the face location.

These results challenge the current location of the touchpad at consumer VR HMDs such as the GearVR which placed its touchpad at the side. The current concept for the GearVR only uses the touchpad for indirect interaction (e.g. swipes). If this would be extended to allow direct touch the positioning should be reconsidered.

**APPLICATIONS**

To present the advantages, explore the design space of display-fixed UIs and show that FaceTouch is also capable of being used with world-fixed UIs we implemented three example applications (cf. video figure). First, we are going to present a general UI concept which we used to embed FaceTouch into VR applications. Afterwards, we present three example applications (gaming controls, text input and 3D modeling) we developed to show how FaceTouch can enhance interaction for current VR applications.

**General UI Concept**

In consumer VR there are currently very little UI concepts to control the device at a general UI level (e.g. control settings inside an IVE). Most devices such as the Oculus Rift and Google Cardboard let the user select applications and content and only afterwards the user puts on the device and immerses into the scene. To change settings the user has to take of the HMD and change those. The reason of which is that VR requires new interaction paradigms incompatible to standard interfaces.

By allowing the control of display-fixed UIs, FaceTouch enables a new way of navigation through UIs in IVES without having to leave the current scene (Fig. 11). The virtual plane can be used to place UI elements similar to current smart phones (e.g. Android). By swiping up and down users can navigate through different virtual planes containing features such as Camera Passthrough, Application Plane or Settings Plane (Fig. 11). Swiping right and left offers settings or further details to the currently selected virtual plane. This allows for interaction with display-fixed UIs without having to leave the current IVE. Since this interaction is not time critical, LiftOff or PressOn can be used as the commit method.

**Gaming Controls**

Games that require the user to control gaze and actions independently from each other (e.g. walking whilst looking around) currently demand to be used with a game controller. Using FaceTouch in combination with LandOn, simple controller elements can be arranged on the touchpad (Fig. 12). LandOn seems most suitable for this application, as it delivered the shortest input times while still providing the low accuracy that this type of application requires. In our implementation of a zombie shooter game we arranged five buttons (four buttons for walking and one for shooting) in a cross over the full touch plane of FaceTouch. The accuracy of the touches is completely sufficient since users don’t have to move their fingers over a great distance but mostly hover over the last touch point (resting the hand on the edges of FaceTouch). This allowed users to control movements independent from the gaze without having to carry around additional accessories.

**Text Input**

Current implementations of applications which need to search through a collection of data (e.g. 360° video databases) on mobile VR HMDs, require the user to browse through the whole library
to find a certain entry. We implemented a simple QWERTY keyboard to input text inside an IVE. Using display-fixed UIs, allows for implementing the keyboard without having to leave the IVE (Fig. 13). Since this scenario requires a precise interaction we used LiftOff as the commit method. In an informal user study we let three experts without training input text (“the quick brown fox...”) resulting in approximately 10 words per minute. This shows the potential of FaceTouch for text input in IVEs, which of course needs further investigation.

3D Modeling
FaceTouch allows not only to select a certain object in 3D space but to rotate, resize, and translate the object by using multi-touch gestures. We implemented a simple “sandbox” 3D modeling application to show the capabilities of FaceTouch. For this application we used the general UI concept which we presented beforehand. Initially the user starts in a blank environment with their touches visualized. Pushing down on the touchpad (PressOn) the user can spawn cubes inside the 3D world. After selecting one cube (PressOn), it can be resized using two fingers (pinch-to-zoom) or rotated using three fingers. By swiping down over the whole touchplane (using three fingers) the user can open a virtual plane showing some control buttons (Fig. 14 right). The user can either fly around the model (movement controls) or select the axis he wants to manipulate (e.g. rotate around x-axis).

LIMITATIONS AND FUTURE WORK
One limitation of the current implementation of FaceTouch is the weight the prototype puts on the user’s head (≈800g). This can be addressed in future prototypes by using more lightweight components. Furthermore, the interaction with a touchpad on the user’s face leads to arm fatigue after a while (similar to the current touchpad at the side of the HMD) which can be counterfeited by supporting the arm and sitting in a comfortable position.

In the future we are planning to enhance the interaction with FaceTouch for multi-touch and two-handed interaction (e.g. for text entry), further investigating the performance. Furthermore, we are planning to explore how gestural interaction can be further embedded into the concept of FaceTouch.

CONCLUSION
Our initial goal of this work was to create an interaction concept which, against the current trend in VR research, focuses on performance for input and not immersion (such as the Leap Motion). We envision touch to become a crucial input method in the future of mobile VR after the first run on “natural” interaction will wear off and people demand a more comfortable form of interaction on a daily basis (or for scenarios where the level of immersion is not essential such as navigating through a menu or even a virtual desktop). We therefore designed FaceTouch to fit into the demand of future mobile VR applications such as quick access to pointing interaction for navigating menus and furthermore the possibility to detach the touchpad and use it in the hands for a longer interaction.

In this paper we presented the novel concept of FaceTouch to enable touch input interaction on mobile VR HMDs. We have demonstrated the viability of FaceTouch for display-fixed UIs using LiftOff for precise interactions such as text entry and LandOn for fast interactions such as game controllers. Our first user study, besides very positive user feedback, revealed important insights into the design aspects of FaceTouch like the right plane distance (MidPlane), impacts of various input methods (LandOn, LiftOff, PressOn) and resulting overshooting behavior. Further we provided optimal target sizes for implementing UIs for LandOn interaction.

Our second user study compared the mounting position for the touchpad and their impact onto the performance of the interaction. We showed that mounting the touchpad on the face resulted in a significant lower error rate for LandOn (8% less than hand and 29% less than side) and LiftOff (2% less than hand and side) and the fastest interaction (LandOn 96 s and LiftOff 1.78 s). The concept of FaceTouch can be furthermore enhanced to also support the ability of removing the touchpad from the mounting position and holding it in the hand. By analyzing the touch behavior of users for all positions we give an indicator of how to implement the targets in terms of size and location.

More importantly, FaceTouch can be combined with other input techniques to further enrich the input space as has been exemplified by the 3D modeling application. Finally, we demonstrated the large design space of FaceTouch by implementing three example applications emphasizing on the advantages of FaceTouch. As FaceTouch can easily be implemented into current mobile VR HMDs such as the Samsung GearVR, we suggest deploying it in addition to HeadRotation. Thereby, for the first time, FaceTouch enables display-fixed UIs as general UI concept (e.g. for text input and menu selection) for mobile VR as well as combined display-fixed UI and world-fixed UI interaction for a much richer experience.

ACKNOWLEDGMENTS
This work was conducted within SFB/TRR 62 Companion-Technology for Cognitive Technical Systems and the Emmy Noether research group Mobile Interaction with Pervasive User Interface both funded by the German Research Foundation (DFG).
REFERENCES


