FaceTouch: Touch Interaction for Mobile Virtual Reality

Jan Gugenheimer
Ulm University
89069 Ulm,Germany
jan.gugenheimer@uni-ulm.de

Gabriel Haas
Ulm University
89069 Ulm,Germany
gabriel.haas@uni-ulm.de

David Dobbelstein
Ulm University
89069 Ulm,Germany
david.dobbelstein@uni-ulm.de

Enrico Rukzio
Ulm University
89069 Ulm,Germany
enrico.rukzio@uni-ulm.de

Christian Winkler
Ulm University
89069 Ulm,Germany
christian.winkler4@uni-ulm.de

CHI'16 Extended Abstracts, May 07-12, 2016, San Jose, CA, USA
ACM 978-1-4503-4082-3/16/05.
http://dx.doi.org/10.1145/2851581.2890242

Abstract
We present *FaceTouch*, a mobile Virtual Reality (VR) head-mounted display (HMD) that leverages the backside as a touch-sensitive surface. *FaceTouch* allows the user to 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 a rich interaction (e.g. gestures) in mobile and nomadic scenarios without having to carry additional accessories (e.g. gamepad). We built a prototype of *FaceTouch* and present interaction techniques and three example applications that leverage the *FaceTouch* design space.

Author Keywords
VR interaction; Mobile VR; VR touch input

ACM Classification Keywords
H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous

Introduction
Virtual Reality (VR) head-mounted displays (HMD) are having a consumer revival with several major companies such as Facebook, Sony and Samsung working on developing consumer devices. Parallel to the devices that are operated by a computer (such as OculusRift and HTC Vive), HMDs have been presented which are operated solely by a mobile phone

Figure 1: The *FaceTouch* prototype. A capacitive touchpad is embedded into a 3D-printed case and attached to the backside of an Oculus Rift DK2 via 5 small buttons that allow for pressure sensing on the touchpad.

(e.g. Samsung GearVR). These mobile VR HMDs allow for new usage scenarios where users can access Immersive Virtual Environments (IVEs) anywhere they want.

These mobile HMDs are expected to penetrate the consumer market more easily, since they only require a smartphone to operate and are currently more affordable (e.g. Google CardBoard). 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 mobile scenarios.

Concept and Implementation

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.

We built a hardware prototype of *FaceTouch* by mounting a 7 inch capacitive touchpad to the backside of a Oculus Rift DK2 (see Fig. 1). 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. 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, as well as a second small touchpad at the side of the device that is used as a comparison in the second user study. 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.

Related Work

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

In order to eliminate finger occlusion during touch interaction, researchers proposed back-of-device interaction (e.g. [6]) which leverages the backside of a mobile device as an input surface. 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 [6]. The human capability of knowing the position and relation of the own body and its several body parts in space is called proprioception [2]. 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 [7]. Further, the use of proprioception was often explored in IVEs [4, 3]. Mine at al. showed the benefits of proprioception in IVEs by letting participants interact with physical props in the non-dominant hand [4]. *FaceTouch* fits into the context of *egocentric* interaction concepts of which the most prevalent are the virtual hand and virtual pointer metaphors [1, 5]. 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 [3]. *FaceTouch* offers the same advantages in terms

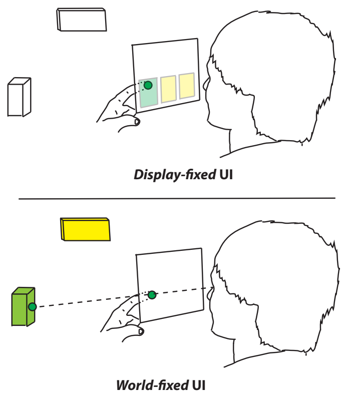

Figure 2: User interface elements for *FaceTouch* can be fixed to both: the *display* (left) and the *world* (right). The virtual plane has a 1:1 direct mapping to the physical touch surface. By touching this plane, users can select *display-fixed* elements on the virtual plane (left) and ray-cast into the scene to select *world-fixed* elements (right).

of passive haptic feedback without forcing the user to hold a physical proxy. *FaceTouch* extends the field by being the first work utilizing back-of-device interaction in VR.

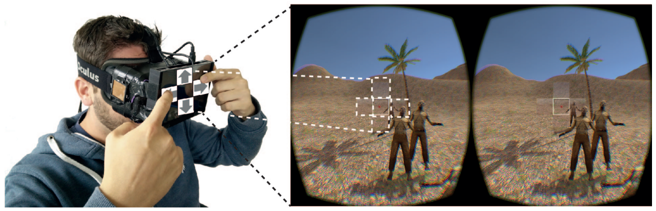

Figure 3: A user controls a first person zombie shooter using *FaceTouch* in combination with *LandOn*. Five buttons for the interaction were arranged in a cross over the full touchpad. This allows for decoupling gaze from interactions such as walking.

Applications

To present the advantages and explore the design space of *display-fixed* UIs and *world-fixed* UIs in combination with *FaceTouch* we implemented three example applications (cf. video figure). These have been tested informally with people at our institution, which is to what described user experiences in the following will refer.

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. 3). 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*. Initially we used arrows to indicate the directions, however users quickly learned the few actions and did not require detailed representations. Therefore, we added a smaller representation of the movement cross only as an indicator. 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 the Oculus Rift).

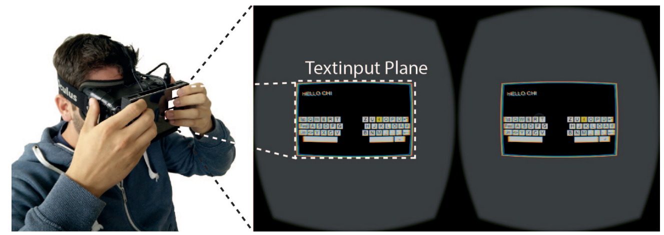

Figure 4: A user is typing text using *FaceTouch* in combination with *LiftOff*. The keyboard is split in half to support the hand posture which is resting at the HMD case.

Text Input

Current implementations of applications which need to search through a collection of data (e.g. 360° video databases) on the Samsung GearVR, require the user to swipe 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. 4). 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" application to show the capabilities of *FaceTouch*.

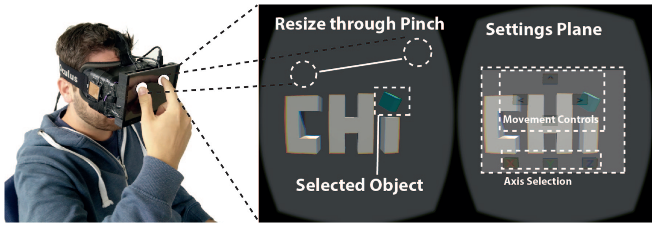

Figure 5: A user creating a 3D model of a CHI logo.
The currently selected object is highlighted in a different color. A pinch gestures is used to resize the currently selected cube. The right eye shows a settings plane which can be opened using a swipe gesture

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. The cubes are currently arranged at one fixed depth position in space. 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. 5 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).

Conclusion

In this paper we presented the novel concept of *FaceTouch* to enable touch input interaction on mobile VR HMDs. We explore the large design space of *FaceTouch* by implementing three example applications emphasizing 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*. In the future we are planing to conduct extensive user studies to evaluate the precision of the interaction.

Acknowledgements

This work was conducted within the Transregional Collaborative Research Centre SFB/TRR 62 "Companion-Technology of Cognitive Technical Systems" funded by the German Research Foundation (DFG).

References

[1] ARGELAGUET, F., AND ANDUJAR, C. A survey of 3d object selection techniques for virtual environments. *Computers & Graphics 37*, 3 (2013), 121–136.

[2] BOFF, K. R., KAUFMAN, L., AND THOMAS, J. P. Handbook of perception and human performance.

[3] LINDEMAN, R. W., SIBERT, J. L., AND HAHN, J. K. Towards usable vr: An empirical study of user interfaces for immersive virtual environments. In *In Proc. CHI '99* (New York, NY, USA, 1999), ACM, pp. 64–71.

[4] MINE, M. R., BROOKS, JR., F. P., AND SEQUIN, C. H. Moving objects in space: Exploiting proprioception in virtual-environment interaction. In *In Proc. SIGGRAPH '97* (New York, NY, USA, 1997), ACM Press/Addison-Wesley Publishing Co., pp. 19–26.

[5] POUPYREV, I., ICHIKAWA, T., WEGHORST, S., AND BILLINGHURST, M. Egocentric object manipulation in virtual environments: Empirical evaluation of interaction techniques. *Computer Graphics Forum 17*, 3 (1998), 41–52.

[6] WIGDOR, D., FORLINES, C., BAUDISCH, P., BARNWELL, J., AND SHEN, C. Lucid touch: A see-through mobile device. In *In Proc. UIST '07* (New York, NY, USA, 2007), ACM, pp. 269–278.

[7] WOLF, K., MÜLLER-TOMFELDE, C., CHENG, K., AND WECHSUNG, I. Does proprioception guide back-of-device pointing as well as vision? In *In Proc. CHI EA '12* (New York, NY, USA, 2012), CHI EA '12, ACM, pp. 1739–1744.