ReconViguRation: Reconfiguring Physical Keyboards in Virtual Reality

Daniel Schneider[1 *] Alexander Otte[1†] Travis Gesslein[1‡] Philipp Gagel[1§] Bastian Kuth[1¶] Mohamad Sham Damlakhi[1‖] Oliver Dietz[1**] Eyal Ofek[2††] Michel Pahud[2‡‡] Per Ola Kristensson[3*] Jörg Müller[4†] Jens Grubert[1‡]

[1]Coburg University of Applied Sciences and Arts [2]Microsoft Research
[3]University of Cambridge [4]University of Bayreuth

Figure 1: Example for reconfiguring the input and output space of individual keys of a physical keyboard in virtual reality. Top row from left to right: emoji entry, special characters, secure password entry using randomized keys. Bottom row from left to right: foreign languages, browser shortcuts, text processing macros.

ABSTRACT

Physical keyboards are common peripherals for personal computers and are efficient standard text entry devices. Recent research has investigated how physical keyboards can be used in immersive head-mounted display-based Virtual Reality (VR). So far, the physical layout of keyboards has typically been transplanted into VR for replicating typing experiences in a standard desktop environment. In this paper, we explore how to fully leverage the immersiveness of VR to change the input and output characteristics of physical keyboard interaction within a VR environment. This allows individual physical keys to be reconfigured to the same or different actions and visual output to be distributed in various ways across the VR representation of the keyboard. We explore a set of input and output mappings for reconfiguring the virtual presentation of physical keyboards and probe the resulting design space by specifically designing, implementing and evaluating nine VR-relevant applications: emojis, languages and special characters, application shortcuts, virtual text processing macros, a window manager, a photo browser, a whack-a-mole game, secure password entry and a virtual touch bar. We investigate the feasibility of the applications in a user study with 20 participants and find that, among other things, they are usable in VR. We discuss the limitations and possibilities of remapping the input and output characteristics of physical keyboards in VR based on empirical findings and analysis and suggest future research directions in this area.

Index Terms: H.5.2: [User Interfaces - Input devices and strategies.]

1 INTRODUCTION

Physical keyboards are common input peripherals, used for tasks ranging from text entry and editing to controlling PC games. Although there is a prevalence of touch keyboards in the mobile world, physical keyboards are still the best tool for text entry, rendering satisfying haptic feedback to the user and enabling fast interaction, even without a visual view of the keyboard [16].

Tracking users' hands and fingers allows the system to visualize the location of the fingers or hands in relation to the keyboard (e.g. [30, 31]). With the recent advances of hand and finger tracking technology (such as HoloLens 2 and Leap Motion), the industry is getting to a point where the use of physical keyboards in VR everywhere, such as in tiny touch-down spaces, is realizable without external tracking. This opens up new opportunities for pervasive VR applications, such as the VR office [29], and motivates research in user interaction with physical keyboards in virtual reality.

Recent research has demonstrated the superiority of physical keyboard for text entry in virtual applications, where the user is practically blind to the real world (e.g., [31,40,55]). Most of these works represented the keyboard by a virtual model of similar geometry, carefully fit to lie at the exact location as the real keyboard. VR is not compelled to follow the physical rules of our real world. Recent research has indicated that the location of the virtual and physical

*e-mail: daniel.schneider@hs-coburg.de. The first three authors contributed equally to this work.
†e-mail: alexander.otte@hs-coburg.de
‡e-mail: travis.gesslein@hs-coburg.de
§e-mail: philipp.gagel@stud.hs-coburg.de
¶e-mail: bastian.kuth@stud.hs-coburg.de
‖e-mail: Mohamad-Shahm.Damlakhi@stud.hs-coburg.de
**e-mail: oliver.dietz@stud.hs-coburg.de
††e-mail: eyalofek@microsoft.com
‡‡e-mail: mpahud@microsoft.com
*e-mail: pok21@cam.ac.uk
†e-mail: joerg.mueller@uni-bayreuth.de
‡e-mail: jens.grubert@hs-coburg.de

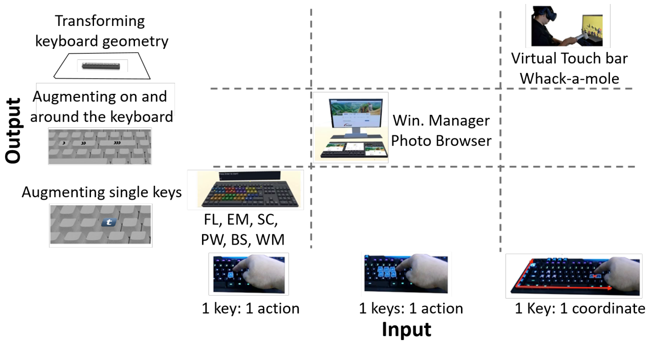

Figure 2: Input-output dimensions of reconfiguring physical keyboards in virtual reality with mapped example applications. The x-axis shows input mappings and the y-axis shows output mappings. FL: Foreign Languages; EM: Emojis; SC: Special Characters; PW: Secure Password Entry; BS: Browser Shortcuts; WM = Word Macros.

keyboard can differ without affecting text entry performance [31]. However, apart from changing the location of the virtual keyboard, further modifications of its visual representation or functionality has not been thoroughly investigated within VR.

In contrast, outside the domain of VR, the idea of repurposing the physical keyboard for more than plain text entry has sparked several research projects regarding how to extend sensing capabilities of keyboards (e.g. [73]) or on how to change the visual appearance of keys (e.g. [5]). We see great potential of taking the idea of reconfiguring physical keyboards and apply it into the domain of VR. Hence, this paper examines how to reconfigure the visual representation and functionality of the keyboard itself. In VR, the keyboard can easily display different graphical elements on its keys and change the number, shape and size of the keys, and enable completely new functionally, such as 1D sliders and image maps. To guide the exploration of different keyboard usages, we introduce *input* and *output mappings*, see Figure 2.

Our contributions in this paper are as follows. We transplant the idea of reconfiguring a physical keyboard to virtual reality and explore a set of possible input and output mappings by designing and implementing nine VR applications that we evaluate in a study. We focus on widely available standard physical keyboards, and, hence, do not investigate the possible input space on and above the keyboard [54]. We investigate the feasibility of the applications in a user study (N=20) and find that they are usable in VR. For two applications (password entry, virtual touch bar) investigated in depth, we find the following: 1) Our results indicate that for password entry shuffling keys in a local region is sufficient for secure password entry; 2) For a selection-based task on a virtual touch bar, changing the visual representation of the keyboard does not affect task performance, however, it has side effects on user's perceived usability and on accidental collisions with the keyboard. We then discuss the limitations and possibilities of remapping the input and output characteristics of physical keyboards in VR based on our empirical findings and analysis and suggest future research directions in this area.

2 Related Work

Our work touches on the areas of physical keyboards in VR, augmenting physical keyboards outside of VR, passive haptics, and security in mixed reality.

2.1 Physical Keyboards for VR

While various modalities can be utilized for text entry in Virtual Reality (for a survey see [18]), recent research has focused on the feasibility of typing on physical full-sized keyboards in VR. An obvious problem is the lack of visual feedback. Without visual feedback users' typing performance degraded substantially. However, by blending video of the user's hands into VR the adverse performance differential significantly reduced [55]. Fundamentally, there are three solution strategies for supporting keyboards in VR. First, by providing complete visual feedback by blending the user's hands into VR. Second, by decoding (auto-correcting) the user's typing to compensate for noise induced by the lack of feedback. Third, by investigating hybrid approaches, such as minimal visual feedback, which may or may not require a decoder to compensate for any noise induced by the method.

Walker et al. [78, 79] investigated typing on a physical keyboard with no visual feedback. They found that the character error rate (CER) was unacceptably high but could be reduced to an average 3.5% CER using an auto-correcting decoder.McGill et al. [55] investigated typing on a physical keyboard in Augmented Virtuality [56]. Specifically, they compared a full keyboard view in reality with a no keyboard condition, a partial and full blending condition. For the blending conditions the authors added a camera view of a partial or full scene into the virtual environment as a billboard without depth cues. They found, that providing a view of the keyboard (partial or full blending) has a positive effect on typing performance. Their implementation is restricted to typing with a monoscopic view of the keyboard and hands and the visualization of hand movements is bound by the update rate of the employed camera (typically 30 Hz). Similar approaches have been proposed [28] and commercialized [6, 77]. Lin et al. [48] investigated the effects of different keyboard representations on user performance and preference for typing in VR but did not study different hand representations in depth.

Grubert et al. [30, 31] investigated the performance of physical and touch keyboards and physical/virtual co-location for VR text entry. They proposed to use minimalistic fingertip rendering as hand representation and indicated that this representation is as efficient as a video-see through of the user's physical hands [30]. Subsequently, similar studies have investigated further hand representations, such as semi-transparent hand models [40]. Besides, optical outside-in tracking systems with sub-millimeter accuracy such as Optitrack Prime series (e.g., in [30, 31, 40]), also commodity tracking devices such as the Leap Motion have been utilized [35] for hand tracking on physical keyboards. Our work extends these previous works by investigating how different input and output mapping on physical keyboards can be utilized in VR for tasks beyond text entry.

2.2 Augmenting Physical Keyboards

There have been several approaches in extending the basic input capabilities of physical keyboard beyond individual button presses. Specifically, input on, above and around the keyboard surface have been proposed using acoustic [38, 43], pressure [17, 52, 81], proximity [74], capacitive sensors [5, 21, 32, 64, 71, 76], cameras [39, 62, 80], body-worn orientation sensors [9] or even unmodified physical keyboards [46, 82]. Besides sensing, actuation of keys has also been explored [4].

Embedding capacitive sensing into keyboards has been studied by various researchers. It lends itself to detect finger events on and slightly above keys and can be integrated into mass-manufacturing processes. Rekimoto et al. [64] investigated capacitive sensing on a keypad, but not a full keyboard. Habib et al. [32] and Tung et al. [76] proposed to use capacitive sensing embedded into a full physical keyboard to allow touchpad operation on the keyboard surface. Tung et al. [76] developed a classifier to automatically distinguish between text entry and touchpad mode on the keyboard. Shi et al. developed microgestures on capacitive sensing keys [70, 71]. Similarly, Zheng et al. [85, 86] explored various interaction mappings for finger and hand postures. Sekimoro et al. focused on exploring gestural interactions on the space bar [69]. While we acknowledge the power of extending the sensing capabilities of physical keyboards, within this work we concentrate on the possibilities of unmodified standard physical keyboards.

Extending the idea of LCD/OLED-programmable keyboards [37, 60, 63], Block et al. extended the output capabilities of touch-sensitive, capacitive sensing keyboard by using a top-mounted projector [5, 26]. Block et al. demonstrate several applications that also inspired our work but they did not evaluate them in a user study. Several commercial products have also augmented physical keyboards with additional, partly interactive, displays (e.g., Apple Touch Bar [2], Logitech G19 [50], Razer Death-Stalker Ultimate [63]). Our work builds on these ideas of changing the physical representations of individual keys or keyboard areas, iterates on them in VR and extends them to change the visual representation of the whole keyboard.

2.3 Usable Security in Mixed Reality

The idea of usable security [68] has been explored within the domain of Augmented and Virtual Reality [1, 27, 45, 65]. De Guzman et al. provide a recent overview of this growing field [15]. The closest work to ours is by Maiti et al. [53] who explore the use of randomized keyboard layouts on physical keyboards using an optical see-through display. Our work is inspired by this idea, translates it and evaluates it in VR.

2.4 Passive Haptics for VR

Passive haptics uses existing props to provide haptic feedback for virtual objects [49]. The user does not need to wear any physical object, but rather comes into contact with it when interacting with a virtual object. Hinckley et al. [34] seminal work was followed by [8], [13], [22] and may be used to simulate object ranging from hand held ones to walls and terrain [[36], [51].

Using existing props raises several challenges addressed by past works. First, a need of a shape similarity between approximating physical object to an virtual one limits the variation of virtual objects that can be represented by a given physical object. Cheng et. al [12] used redirection of the users' hand to map a variety of virtual geometries to a fixed physical proxy and measure a difference between real and virtual geometries. Zhao and Follmer [84] suggested a parametric optimization to calculate needed retargeting. In an effort to create haptic proxies that are close to the virtual object shapes, researchers printed specific physical props [34] [42]. Shape display devices, are active shape changing devices that try to mimic the virtual object geometry, such as LineForm rendering curves [Nakaguki 15], actuated pins arrays ([24], [41]), and constructive blocks [66], [67]. Teather et al. [75] evaluated the effects of co-location of control and display space in fishtank VR and found only subtle effects of co-location in an object movement task using a tracked stylus. Similarly, further research on visuo-motor co-location on 3D spatial tasks resulted in inconclusive results, not indicating statistically significant differences [23, 25].

In this paper, we do look at an existing physical object: a keyboard. While every surface that is used for touch is represented at the exact location in the virtual space, we do look at ways in which we can modify the keyboard in the virtual world – highlighting new functionalities and obfuscating unused parts of the keyboard.

A second challenge is the need to track the position of the physical proxy in space. Corsten et al. [14] provide the ability to recognize and track these physical props using Kinect. Since we use a dedicated device – a keyboard, we can mark it to ease tracking by the virtual reality system. We may use active trackers such as Vive Tracker [77] or passive visual trackers, as used in this paper.

Finally, there is a challenge to populate the user's physical environment with a range of passive objects that may be used as proxies for a slew of virtual objects. Hettiarachchi et al. opportunistically repurpose physical objects to provide haptic responses to specific widget-like digital controls, such as sliders and buttons. Annexing reality [33] analyzes the environment and opportunistically assigns objects as passive proxies. Azmandian et al. [3] allow the use of one such passive proxy for several virtual ones in the same virtual space, yet the problem of finding a close enough proxy in the environment is required. In this work an existent of a physical keyboard in the user's vicinity is assumed.

3 Reconfiguring Physical Keyboards in VR

The fundamental dimensions that directly tie into physical keyboard reconfiguration are its *input mapping* and *output mapping*, see Figure 2. The first input mapping is a direct mapping of a physical key to an action, which effectively results in the key being overloaded. Examples include any method that reconfigures key labels and their corresponding actions, such as displaying emoticons or special characters on the keys and then outputting the corresponding emoticon or symbol. The second input mapping is to map multiple physical keys to the same single action. This allows a section of the keyboard to correspond to the same user interface element. For example, a user interface element such as a photo, where parts of the photo are mapped to different keys but selecting any such key trigger the same action, such as selecting the photo. This can be convenient when there is not enough space on a single key to display a photo or an application that could be selected, or when we want to make a set of keys stand out as a larger key. The third input mapping uses an area of the keyboard as a map where each single physical key is mapped to a single coordinate in the map.

We consider three characteristics emerging from the output dimension. First, output can be revealed by augmenting a single key in VR. This allows individual keys to take on specific visual functions, such as for example each key representing a distinct emoticon or special character. Second, output can be generated by augmenting the visual display on and around the keyboard. For example, several keys (and the spaces in between them) can be mapped to displaying a single user interface element, such as a photo. Alternatively, the entire keyboard and its surrounding can be reappropriated as an integrated environment with its individual interface elements mapped to a single or multiple keys. An example of this is the whack-a-mole game shown in Figure 4. Third, the keyboard geometry itself can be transformed into a user interface control with different characteristics, such as an output design of part of the keyboard that resembles a continuous 1D slider. While selection on the slider is discrete due to the input modality still consisting of individual physical keys, the output modality appears continuous to the user.

3.1 Applications

Clearly it is impossible to exhaustively explore the rich input and output mappings that are realizable when allowing physical keyboards to be reconfigured for VR. Instead, we probed the input-output dimensions by selecting a diverse set of VR applications within the input and output mappings shown in Figure 2. The applications were selected by considering several factors that collectively result in a diverse set of viable and interesting VR applications: the type of computer tasks people are engaging with today and how they can be improved upon or supported in VR, the degree of input keyboard reconfigurability required, the level of visual output design inside and around the keyboard, the task complexity, and objective of the user (such as office work, rapid access to functionality, window multitask support and entertainment).

3.1.1 Emojis

Emojis, i.e., Unicode graphic symbols, used as a shorthand to express concepts and ideas are increasingly used in mobile scenarios [58]. To support efficient emoji entry using a physical keyboard, we enabled emoji entry through pressing a modifier key (e.g., CTRL)

Figure 3: Top row from left to right: Window manager with three virtual buttons, photo browser with 24 and 104 images, view on virtual monitor with selected image. Bottom row from left to right: Virtual touchbar for controlling the timeline of a movie with highlighted keys, as one row keyboard plus red bounding box, without any visualized key, view on the virtual monitor with selected frame.

Figure 4: Virtual Whack-A-Mole. Left: User hammering on the keyboard with a spatially-tracked prop. Right: view on the virtual scene.

that switches the normal keyboard mapping to an emoji set, see Figure 1, top row, left. To support an increasing number of emojis, the emoji set can be changed by pressing the modifier key and the page up / down buttons. We will refer to this application as EMOJIS in the remainder of this paper.

3.1.2 Languages and Special Characters

For those who write in multiple languages, the respective character mappings can be visualized on the virtual keyboard, see Figure 1, bottom row, left, for Cyrillic characters. Supporting multiple languages on keyboard is becoming very popular in mobile scenarios with touch keyboards. We implemented language mappings for Arabic, Cyrillic, Greek, Hindi and Japanese. Languages can be switched by pressing a modifier key and the page up / down buttons.

Further, special characters (such as umlauts) can be entered by first pressing a modifier key (in this case, ALT) and the respective letter (e.g., ”a”) using one hand. Then the neighboring keys are highlighted in green and will temporarily show the available umlauts, see Figure 1, top row, middle. The umlaut can then be selected by pressing the respective green key with a third finger (e.g., using the second hand). We will refer to this application as LANGUAGES in the remainder of this paper.

3.1.3 Application Shortcuts

Keyboard shortcuts for applications have the benefit of triggering actions fast but at the cost of memorizing these shortcuts, which can become challenging with shortcuts reaching dozens to hundreds in applications such as Adobe Photosop or Microsoft Office [44, 59].

In order to support the keyboard shortcut discoverability, we implemented shortcuts for a virtual Web-browser (based on Zenfulcrum[1]), see Figure 1, bottom row, middle. Specifically, we implemented navigation shortcuts (back, forth, home), refresh, cancel and access to 10 bookmarked webpages. We will refer to this application as BROWSERSHORTCUTS in the remainder of this paper.

[1]https://zenfulcrum.com/, last accessed March 19, 2019.

3.1.4 Virtual Text Processing Macros

While keyboard shortcuts are often reflecting a predefined set of actions, new application actions can also be defined. To this end, we defined macros in Microsoft Word (insert signature, insert sender's address, insert image) and mapped them to individual keys (again triggered by a modifier key) on the virtual keyboard. Inclusion of desktop applications such as Word can be achieved by utilizing a virtual desktop mirror[2]. We will refer to this application as WORD-MACROS in the remainder of this paper.

3.1.5 Window Manager

Various schemes for switching open applications (e.g., using ALT-TAB on Windows) have been proposed for Window managers. In VR, it is possible to visualize open applications directly on the keyboard, see Figure 3, top row, left. Alternative selection (to alt-tabbing or mouse-based selection), can be achieved by pressing one key in the area below the visualized open window. We will refer to this application as WINDOWMANAGER in the remainder of this paper.

3.1.6 Photo Browser

Similarly, to switching open windows, browsing and selecting photos can be achieved. In a virtual photo browsing application a set of available photos is mapped to the keyboard, see Figure 3, top row, second and third from left to right. The number of photos to be visualized at once is limited by the number of physical keys on the keyboard (e.g., 104 or 105). Similarly, to the emoji application, further photos can be shown when buttons (such as page up / down) are reserved for toggling between photo sets. We will refer to this application as PHOTOBROWSER in the remainder of this paper.

3.1.7 Whack-a-Mole

Besides productivity tasks, the keyboard can also be utilized for leisure applications. We implemented a whack-a-mole game by turning the keyboard into a virtual playground. Moles are digging there way through the ground and have to be whacked with a virtual hammer. To this end, we utilized a physical hammer prop, that is spatially tracked, see Figure 4, but also implemented a version that does not rely on a physical prop (basically triggering a hammer movement when a keyboard area is hit by the user's hand). We will refer to this application as WHACKAMOLE in the remainder of this paper.

The Whack-a-Mole is just one example of usage of the keyboard surface as an image map, and enabling a very quick accessing a map point by pressing a key. Although this application is a game and uses a hammer, we could also imagine dynamic modifications

[2]https://github.com/Clodo76/vr-desktop-mirror, last accessed March 19, 2019

of visuals on or over the keyboard used for learning applications. With WHACKAMOLE, we display moles and ask the user the react quickly. As an example, this concept could translate to typing applications, where we could highlight which key(s) should be tapped with which finger(s) of which hand and measure the user's reaction time. In addition, for beginner typists, we could also display over the keyboard a semi-transparent 3D model of a hand to explain how the hand should approach the keyboard for each key as the user is typing.

3.1.8 Secure Password Entry

When engaging with a physical keyboard while wearing a immersive HMD, users can become unaware of their surroundings [55]. This can become critical, when users enter sensitive data such as passwords. To support users in minimizing the risks of shoulder surfer attacks [19], we implemented three different key randomization strategies, see Figure 8, which offer varying trade-offs between guessability of the pressed key by a shoulder surfer and discoverability of the key by the VR user. In REGIONSSHUFFLE, keys are randomized in the local region of the key. In ROWSHUFFLE, keys are randomly assigned along the original row. In FULLSHUFFLE, keys are randomly assigned across the keyboard. Please note, that ROWSHUFFLE and FULLSHUFFLE have been proposed in the domain of AR text entry [53]. As a first-order approximation, we estimate the probability p of an observer correctly guessing a password of length n as:

$$p = k^{-n}, \tag{1}$$

where k is the number of keys that are shuffled. This formula assumes 1) the password is truly random; 2) an observer has perfect ability to always correctly infer the key location of any key press of the user; and 3) the observer knows the shuffling system (Kerckhoffs's principle).

While passwords in practice are not truly random, we can use the above estimation to create an illustration of the theoretical trade-off between the effort incurred by the user in searching for the keys on a shuffled keyboard against the probability of an observer inferring the password by observing the user's hand movement. The time T it takes for a user to type a password of length n with k keys shuffled can be estimated as:

$$T = n \cdot KT + (1-\alpha)(n-1)(k \cdot DT) + k \cdot DT, \tag{2}$$

where KT is average time to move to a key and press it, DT is average decision time, that is, the time it takes a user to look at a key and decide whether it is the intended key or not, and $\alpha \in [0,1]$ is a parameter specifying the user's ability to memorize the location of the shuffled keys after an initial scan ($\alpha = 0$: no memory; $\alpha = 1$: perfect memory).

Equations 1 and 2 form a system that captures the trade-off in time required by the user to type the password compared to the probability of an observer inferring the password. KT and DT are empirical parameters that can be held constant while n and k are controllable parameters of the system and α is an uncontrollable parameter that varies according to the individual user's ability to quickly remember the shuffled layout (and depending on how often the system changes the key layout). Thus setting KT and DT to appropriate values and holding n, k and α constant reveals the trade-off between time incurred to type a password and the probability of an observer inferring the password as a function of password length, the number of shuffled keys and the ability of the user to memorize the shuffled layout.

For example, assume the desired probability of an observer at any given instance is able to infer the password is set to one in a million and password length is set to 8. Solving for the integer k in Equation 1 results in $k = \lceil p^{-\frac{1}{n}} \rceil$ and thus the number of keys that should be shuffled is 6. Using nominal values for typing a letter of random text and deciding whether an individual key is the intended one [10] provides estimations of $KT = 0.5$ s and $DT = 0.24$ s. At this operating point, Equation 2 then provides a time estimate of approximately 5.5 s for typing the password if the user has perfect memory of the reshuffled layout after the first scan and 15.5 s if the user has no memory. As Equation 2 is a linear combination, the user's ability to memorize the layout after a scan will linearly affect the time prediction at any given point between the two extremes. We emphasize that we do not make an attempt to introduce an accurate model but merely wish to illustrate that a first-order approximation is sufficient for making the inherent trade-offs of the design parameters explicit to the designer.

3.1.9 Virtual Touch Bar

Inspired from the ideas of LCD/OLED-programmable keyboards [37, 60, 63] and, in recent years, keyboards augmented with additional displays (e.g., Apple Touch Bar [2]) we explore how ordinary physical keyboards can be turned into quasi touch bar keyboards in VR. While a physical keyboard lacks the input resolution of an actual capacitive-sensing touch bar, a physical keyboard in VR has the potential benefit of 3D content visualization. Further, the visualizations are not limited to solely augment the keyboard.

In our prototype, We implemented an application for controlling the seek bar of a virtual video player. In order to increase the input resolution of the virtual touch bar keys (number keys was between 1 and 10), users could press two adjacent buttons to select an intermediate frame, e.g., if pressing key 1 would jump to second 10 in a 100 second film and key 2 would jump to second 20, then pressing keys 1 and 2 simultaneously would jump to second 15.

As a base visualization, we highlighted relevant keys for interaction, as shown in Figure 3, bottom row, left. We will call this VTHIGHLIGHT in the remainder of the paper.

VR allows us to change the visual representation of individual keys or the whole keyboard. To explore this idea, we implemented a one row keyboard that is complemented by a colored bounding box that indicates the geometric bounds of the whole keyboard, see Figure 3, bottom row, second from the left). The design rationale behind this choice was to allow users to have an accurate visual representation of the physical keys that are relevant for interaction and, at the same time, to give a visual indication of the physical dimensions of the inactive part of the keyboard. We call this implementation VTONEROW in the remainder of the paper.

A further variation is to completely hide the keyboard and replace it by a relevant graphical user interface representation for the task at hand. To this end, we only indicated a 1D slider element at the place of the original keys (see Figure 3, bottom row, third from left). We call this implementation VTINVISIBLE in the remainder of the paper. Each variant has its own potential benefits and drawbacks. The most apparent design issue is perceived accordance [57]: will users be able to perceive the given functionality given the visual representation? This is explored further in the user study in the next section.

4 USER STUDY

The purpose of our user study was threefold. First, we wanted to get feedback to learn from initial user reactions. To this end, we followed the approach by Chen et al. of demonstrating individual experiences to users instead of carrying out task-based evaluations [11]. To this end, we demonstrated the following applications described in Section 3.1: WHACKAMOLE, PHOTOBROWSER, WINDOWMANAGER, WORDMACROS, BROWSERSHORTCUTS, LANGUAGES, EMOJIS and asked the participants to engage with each application. Second, we wanted to understand how physical keyboards can be utilized to support usable security in the context of shoulder surfing attacks in VR. Similar studies have been conducted in AR [53]. However, it is unclear how password entry using shuffled keys translates to VR due to the different output media (e.g., AR glasses project the keys into a different depth layer than the physical keys). Further, and in contrast to previous work, our focus was to better understand the relation between objective and perceived security as well as the trade-off between perceived security and text entry performance. Third, we wanted to investigate the effects of changing the visual representation of a physical keyboard on user experience and performance. To this end, we employed a selection-based task in context of the virtual touch bar app.

Figure 5: Spheres used as fingertip visualization during the experiment.

4.1 Participants

We recruited 20 participants from a university campus with diverse study backgrounds. All participants were familiar with QWERTZ desktop keyboard typing. From the 20 participants (5 female, 15 male, mean age 27.8 years, $sd = 3.8$, mean height 176.3 cm, $sd = 8.6$), 8 indicated to have never used a VR HMD before, four participants once, four participants rarely but more than once, two participants occasionally and two participant to wear it very frequently. Four participants indicated to not play video games, one once, two rarely, 6 occasionally, four frequently and three very frequently. Seven participants indicated to be highly efficient in typing on a physical keyboard and 12 to write with medium efficiency on a physical keyboard and one to write with low efficiency on a physical keyboard (we caution against over-interpreting these self-assessed performance indications). Nine participants wore contact lenses or glasses. Two volunteers have participated in other VR typing experiments before.

4.2 Apparatus and Materials

An OptiTrack Prime 13 outside-in tracking system was used for spatial tracking of finger tips, the HMD and the keyboard. The tracking system had a mean spatial accuracy of 0.2 mm. A HTC Vive Pro was used as HMD. As physical keyboard a Logitech G810 was used. The setup is shown in Figure 6. We utilized a minimalistic finger tip representation as suggested by prior [30] work to indicate the hand position relative to the keyboard, see Figure 5.

For the passwords we used a set where half of the passwords were popular simple passwords and the other half was split equally to five and 10 character length randomized passwords. The virtual environment was showing the virtual keyboard and a monitor resting on a desk. In line with previous work [30, 31] the passwords and the entered text were visualized both on the monitor and directly above the keyboard. The system was implemented in Unity 2018.2 and deployed on a PC (Intel Xeon E5-1650 processor, 64 GB RAM, Nvidia GTX 1070 graphics card) running Windows 10.

4.3 Procedure and Task

After welcoming, participants were asked to fill out a demographic questionnaire. Thereafter, we conducted a calibration phase to be able to spatially track the fingers of the participants. Retro-reflective markers were fixated with double sided tape on the participants' fingernails, see Figure 7, bottom row.

Figure 6: Apparatus of the experiment showing a participant with nail attached retro-reflective markers, the G810 keyboard, a HTC VIVE Pro Headset, the Optitrack Prime 13 tracking system and an external webcam for logging. Please note that the headphones are not attached to the ears.

The study was divided into three parts with 5-minute breaks in between. The first part gathered user feedback on the seven applications (see below). This part lasted around 30 minutes. After that evaluation of the password entry (lasting ca. 35 minutes) and virtual touch bar (lasting ca. 20 minutes) applications were carried out alternately. Thus, either of those two applications was executed as the second and the other as the third. After each part, a questionnaire was filled out by the participants to rank the conditions. Then a semi-structured interview followed. On average, the procedure took 110 minutes. Finally, the participants were compensated with a 10 Euro voucher. Due to the already long study duration we employed a short three-item questionnaire capturing the user experience dimensions of ease of use, utility and enjoyment (in line with similar approaches [11, 47]), instead of utilizing longer questionnaires such as the 10-item system usability scale [7].

In the first part, we demonstrated the following seven applications to the participants and asked them to engage with the applications themselves: WHACKAMOLE, PHOTOBROWSER, WINDOWMANAGER, WORDMACROS, BROWSERSHORTCUTS, LANGUAGES, EMOJIS. We asked for feedback on ease of use, utility and enjoyment for each application after demonstration. The order of the applications was balanced across participants, as good as possible (full permutation was not possible due to the number of applications). The second part alternated with the third part for counterbalancing across the participants. To evaluate the password entry application the participants were asked to type passwords for two minutes from a predefined set of the most used passwords[3]. Two baselines where taken without the HMD and wearing the HMD. For each of the three counterbalanced conditions (REGIONSHUFFLE, ROWSHUFFLE, FULLSHUFFLE) there was a one minute training phase, followed by the 5 minute testing phase. After each condition a questionnaire with questions about ease of use, utility and enjoyment was answered.

The third part, alternating with the second part, was conducted to get insight into changing the visual representation of the keyboard. To this end, we employed the virtual touch bar application, with the three coutnerbalanced conditions (VTHIGHLIGHT, VTONEROW and VTINVISIBLE), see Figure 7. After showing the virtual touch bar in a demo, the participants were asked to use the touch bar with a repeated motion. Starting at a fixated point in the VR (centered 3 cm below the bottom edge of the keyboard), the participants had to move their index finger of the dominant hand to the touch bar and back to the fixated point (basically following the procedure

[3] https://en.wikipedia.org/wiki/List_of_the_most_common_passwords, last accessed March 21, 2019

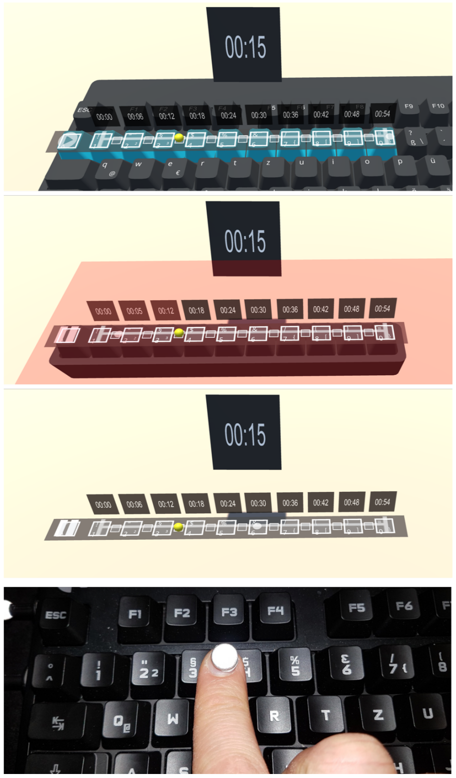

Figure 7: Close-up view on the conditions in the virtual touchbar task. From top to bottom: VTHighlight, VTOneRow, VTInvisible, view on the physical keyboard with a user's finger and attached retro-reflective marker for fingertip tracking.

Figure 8: Conditions in the password entry experiment. From top to bottom: RegionsShuffle, RowShuffle, FullShuffle.

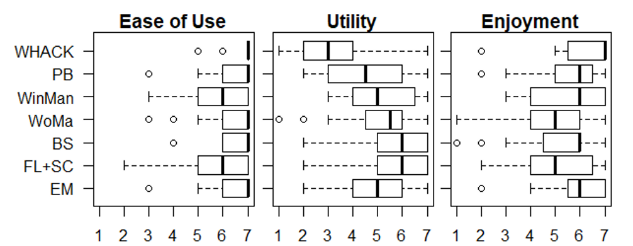

Figure 9: Ease of use ratings, Utility and Enjoyment ratings on a 7-item Likert scale (1: lowest, 7: highest). Abbreviations: WHACK: WhackAMole, PB: PhotoBrowser, WinMan: WindowManager, WoMa: WordMacros, BS: BrowserShortcuts, FL+SC: Languages, EM: Emojis

of a Fitts Law task with fixed target size). The fixated point was connected to the coordinates of the VR-keyboard, to guarantee a static distance to the targets. As a baseline 5 timestamps were shown to the participant, that he had to locate in the touch bar. After this the condition was conducted with 25 timestamps. The conditions were counterbalanced across participants.

4.4 Results

Unless otherwise specified, statistical significance tests for performance data (text entry rate, error rate, task completion time) were carried out using general linear model repeated measures analysis of variance with Holm-Bonferroni adjustments for multiple comparisons at an initial significance level $\alpha = 0.05$. We indicate effect sizes whenever feasible (η_p^2). For subjective feedback, or data that did not follow a normal distribution or could not be transformed to a normal distrubtion using the log-transform, we employed Friedman test with Holm-Bonferroni adjustments for multiple comparisons using Wilcoxon signed-rank tests.

4.5 Initial user feedback on Applications

Figure 9 shows user ratings on seven-item Likert scales for questions on ease of use ("I found the application easy to use"), utility ("I found the application to be useful") and enjoyment ("I had fun interacting with the application"). The figure indicate high ratings for ease of use, varying ratings for utility and for enjoyment. Please note that we did not run null hypothesis significance tests on these ratings as they should serve as a descriptive indication of these user experience dimensions only.

Users were asked to comment on the individual applications. We followed top-down qualitative coding and structuring procedures to identify benefits and drawbacks of the individual applications [72]. For WhackAMole users mentioned that it was "fun and totally different experience", but also that while "easy to understand" it is "not relevant for work" with one participant stating "The PC is a tool for me. I prefer things that make work easier". Regarding both PhotoBrowser and WindowManager users were critical with liking the principal idea but noticing, that when the icons get too small (ca. 4 times the key size), the experience results in "low usability". For WordMacros, opinions were split, with some users mentioning it usefulness and ease of use (e.g., "I found it useful and easy to use") and others questioning its utility ("I do not think this is useful for work"). User's generally appreciated the BrowserShortcuts applications with one mentioning "Shortcuts are important for productivity. It is not bad to see all of them" and another one saying "I need to switch between tabs at work often. It is useful". Regarding foreign languages and special characters in the Languages application, opinions were split for the language mapping according to the cultural background of the users. A user who did not type in multiple languages mentioned explicitly "This is not useful for me. I do not use other languages" and one multi-lingual user mentioned "I find this useful for foreign languages because otherwise it is a lot of work to type special symbols or characters". Regarding special characters opinions were similar, with one user mentioning "I do not use special characters" and another one "I find this useful for formulas and special characters". However, some users also mentioned the unexpected layout of special characters (in contrast to a simple row on soft keyboards) with one stating: "There is much searching required" to find the needed character. Finally, for Emojis, users generally appreciated the application with one user "I would use this in daily life" and another one saying "This is a satisfying way" (to type emojis) "even if not in VR". However, a productivity oriented user also mentioned "Emojis are unnecessary in daily work".

4.6 Password Entry

We report on text entry rate, character error rate, user experience ratings as well as preferences and open comments in the next subsections. In our performance evaluation, we concentrate on the joint set of simple and randomized passwords. While the absolute performance values differ between password sets with respect to text entry speed, the significance between conditions did not change.

4.6.1 Entry Rate

Entry rate was measured in words-per-minute (wpm), with a word defined as five consecutive characters, including spaces. The entry rate measured in the profiling phase without randomized keys was 21.0 wpm ($sd = 7.91$). The mean entry rate for REGIONSHUFFLE was 6.57 wpm ($sd = 1.96$), for ROWSHUFFLE it was 6.03 wpm ($sd = 1.63$) and for FULLSHUFFLE it was 3.82 wpm ($sd = 1.44$).

An omnibus test revealed significance ($F_{3,17} = 49.73$, $\eta_p^2 = 0.898$, $p < .001$). Holm-Bonferroni adjusted post-hoc testing revealed significant differences between baseline and all randomization layouts (which was to be expected) (adjusted p-values < 0.001), between FULLSHUFFLE and REGIONSHUFFLE (adjusted p-value $< .001$) as well as between FULLSHUFFLE and ROWSHUFFLE (adjusted p-value $< .001$), but not between REGIONSHUFFLE and ROWSHUFFLE (adjusted p-value $= 1.00$). In other words, FULLSHUFFLE lead to significantly reduced text entry speed compared to both REGIONSHUFFLE and ROWSHUFFLE.

4.6.2 Error Rate

Error rate was measured as character error rate (CER). CER is the minimum number of character-level insertion, deletion and substitution operations required to transform the response text into the stimulus text, divided by the number of characters in the stimulus text. The CER measured in the profiling phase without randomized keys was 3.2% ($sd = 3.5$). The CER for REGIONSHUFFLE was 3.7% ($sd = 3.7$), for ROWSHUFFLE it was 3.4% ($sd = 3.4$) and for FULLSHUFFLE it was 3.4% ($sd = 5.3$). An omnibus test revealed no significance ($F_{3,5} = 0.55$, $\eta_p^2 = 0.032$, $p < 0.981$). In other words, there where no significant differences in terms of error rate between the conditions.

4.6.3 User Experience Ratings

User ratings regarding ease of use, utility and enjoyment (utilizing the same question as in Section 4.5) are shown in Figure 10. A Friedman test indicated statistically significant differences between the conditions for ease of use ($\tilde{\chi}^2 = 23.52$, $p < .001$), and enjoyment ($\tilde{\chi}^2 = 12.54$, $p = .002$) but not for utility ($\tilde{\chi}^2 = 0.98$, $p = .61$) .

Regarding ease of use, Bonferroni adjusted Wilcoxon signed rank tests indicated pairwise differences between FULLSHUFFLE and REGIONSHUFFLE ($Z = -3.55$, $adjusted p < .001$) as well as between FULLSHUFFLE and ROWSHUFFLE ($Z = -3.43$, $adjusted p = .002$), but not between REGIONSHUFFLE and ROWSHUFFLE ($Z = -0.722$, $adjusted p = .94$). Regarding enjoyment, Bonferroni adjusted Wilcoxon signed rank tests indicated pairwise differences between FULLSHUFFLE and REGIONSHUFFLE ($Z = -2.96$, $adjusted p = .009$) as well as between FULLSHUFFLE and ROWSHUFFLE ($Z = -2.80$, $adjusted p = .015$), but not between REGIONSHUFFLE and ROWSHUFFLE ($Z = -0.05$, $adjusted p = .96$).

We also asked participants to rate the conditions regarding perceived security. The average score on a 7-item Likert scale (1: totally disagree, 7: totally agree) for the statement "I felt protected from shoulder surfers" where 6.10 ($sd = 1.45$) for FULLSHUFFLE, 5.60 ($sd = 1.57$) for REGIONSHUFFLE and 6.05 ($sd = 1.43$) for ROWSHUFFLE. For the statement "I think that the proposed condition makes password entry more secure" the ratings where 6.05 ($sd = 1.28$) for FULLSHUFFLE, 5.50 ($sd = 1.32$) for REGIONSHUFFLE and 5.60 ($sd = 1.39$) for ROWSHUFFLE. While Friedman tests indicated significant differences between conditions, post-hoc comparisons with Bonferroni adjustment failed to indicate pairwise differences. In other words, FULLSHUFFLE led to a significant lower ease of use and enjoyment rating compared to both other conditions.

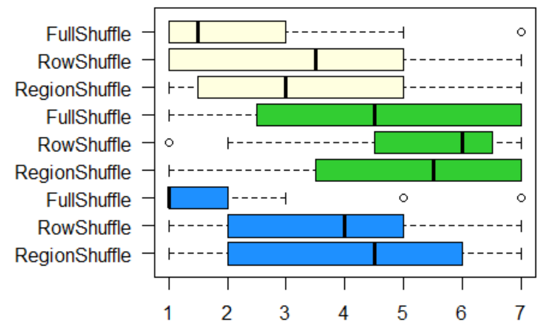

Figure 10: Ease of use (blue), Utility (green) and Enjoyment (yellow) ratings on a 7-item Likert scale (1: lowest, 7: highest).

4.6.4 Preferences and Open Comments

REGIONSHUFFLE was preferred by 9 participants, ROWSHUFFLE by 10 participants and FULLSHUFFLE by one participants. Regarding REGIONSHUFFLE, four users mentioned "It is easier to coordinate.", one mentioned at this to be the "most usable" option and another one commening on the security aspect with saying this option had "enough shuffling". For ROWSHUFFLE four user mentioned characters are " easier to find in the row", one saying "this is closest to normal use". However, regarding perceived security two users mentioned a perceived "low security". For FULLSHUFFLE six users mentioned "it is frustrating" and another six "It is very time consuming". Regarding perceived security five users mentioned "It has the best security".

4.7 Virtual Touch Bar

We report on task completion time, selection errors, collisions with the keyboard, user experience ratings as well as preferences and open comments in the next subsections.

4.7.1 Task Completion Time and Errors

The mean task completion time for VTHIGLIGHT was 2.15 seconds ($sd = 0.43$), for VTONEROW it was 2.15 seconds as well ($sd = 0.42$) and for VTINVISIBLE it was 2.24 seconds ($sd = 0.39$). An omnibus test did not reveal significance ($F_{2,16} = 2.26$, $\eta_p^2 = 0.22$, $p = .14$).

The average number of errors (i.e., users pressed a wrong key) was VTHIGLIGHT was 0.85 ($sd = 1.39$), for VTONEROW it was 1.55 ($sd = 2.92$) and for VTINVISIBLE it was 1.65 ($sd = 3.94$). A Friedman test did not reveal significance ($\tilde{\chi}^2 = 3.13$, $p = .21$). In other words, all visualizations resulted in comparable performance measures.

4.7.2 Collisions

We observed the number of accidental collisions between the user's hand and the physical keyboard through an external camera and an additional human observer. For VTHIGLIGHT the average number of collisions were 0.49 ($sd = 0.82$), for VTONEROW 2.45 ($sd = 2.11$) and for VTINVISIBLE the mean number of collision was 3.65 ($sd = 3.51$).

A Friedman test indicated statistically significant differences between the conditions ($\tilde{\chi}^2 = 21.73$, $p < .001$). Bonferroni adjusted Wilcoxon signed rank test indicated pairwise differences between

VTHiglight and VTOneRow ($Z = -3.20$, $p = .001$) as well as between VTHiglight and VTInvisible ($Z = -3.740$, $p = .000$), but not between VTInvisible and VTOneRow ($Z = -1.24$, $p = .22$). In other words, showing the full keyboard inVTHiglight led to a significant reduced number of collisions compared to both other visualizations.

4.7.3 User Experience Ratings

User ratings regarding ease of use, utility and enjoyment (utilizing the same question as in Section 4.5) are shown in Figure 11. Friedman tests did not reveal significant differences.

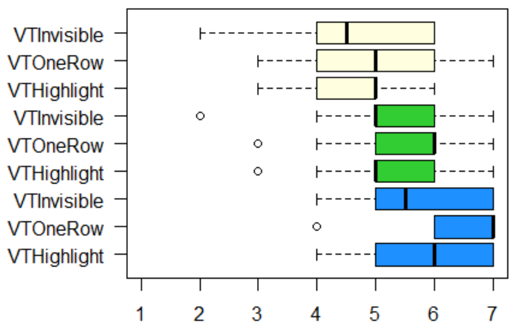

Figure 11: Ease of use (blue), Utility (green) and Enjoyment (yellow) ratings on a 7-item Likert scale (1: lowest, 7: highest).

We also asked participants to rate the conditions regarding the following statement "I had the feeling I could hit the targets that I was aiming for" The average score on a 7-item Likert scale (1: totally disagree, 7: totally agree) was 6.30 ($sd = 1.13$) for VTHighlight, 6.15 ($sd = 1.23$) for VTOneRow and 5.53 ($sd = 1.17$) for VTInvisible. A Friedman test indicated statistically significant differences between the conditions ($\tilde{\chi}^2 = 12.94$, $p = .002$). Bonferroni adjusted Wilcoxon signed rank test indicated pairwise differences between VTInvisible and VTHiglight ($Z = -2.81$, $adjusted p = .015$) as well as between VTInvisible and VTOneRow ($Z = -2.84$, $adjusted p = .015$), but not between VTHighlight and VTOneRow ($Z = -.91$, $adjusted p = 1.0$).

4.7.4 Preferences and Open Comments

VTHighlight was preferred by 11 participants, VTOneRow by six participants and VTInvisible by three participants.

For VTHighlight five users mentioned "I have better orientation" (compared to the other representations), three saying "I have a better feeling where my fingers are", two that this representation "feels most natural" but also two mentioning that there where "too many keys" that "are confusing".

Regarding VTOneRow two participants mentioned that this represenation "has enough information to navigate", but also one user mentioning that it "feels redundant because of the other two options"

Finally, for VTInvisible a user said "it gives a good overview without useless information". However, six users mentioned "It is hard to navigate correctly" and another one "It is hard to handle".

5 Discussion

The first part of the study indicated that the proposed seven applications WhackAMole, PhotoBrowser, WindowManager, WordMacros, BrowserShortcuts, Languages, Emojis were mostly usable but varied in the utility rating, partly based on the participants' background (e.g., uni- vs. multilingual). However, for both PhotoBrowser and WindowManager participants indicated a reduced usability, when the visualized images approach the size of individual keys. This indicates, that the input resolution of the keyboard was higher than what was usable for those two applications (which utilized screenshots of applications or websites for visualization). In contrast, for other applications using small symbols (such as Emojis, BrowserShortcuts) no remarks regarding legibility of symbols were made (potentially due to the better discernability of those graphic symbols).

5.1 Password Entry

The study results for password entry indicated that shuffling the keys over the whole keyboard in condition FullShuffle lead to a significant lower text entry rate (which is to be expected) compared to the other two shuffling schemes and, at the same time, to significantly lower user experience ratings (in terms of ease of use and enjoyment), which is also reflected in the user comments (e.g., "it is frustrating"). Interestingly, while five users explicitly mentioned that FullShuffle "has the best security" the results from the perceived security questions do not fully support this hypothesis due to a lack of statistical significance. Given the current evidence, we would, hence, argue that regional shuffling of keys seems to be sufficent for future usage as it leads to less user frustration without significantly scarifying perceived security. The first-order model we introduced earlier in this paper would predict a password entry rate for shuffling six keys that lie approximately between 0.5 and 1.5 characters-per-second (cps), which corresponds to between 6 and 18 wpm which roughly correspond to the lower-end to the entry rate observed in the user study.

5.2 Virtual Touch Bar

The experiment indicated that the visual representation of the keyboard can be changed without a significant impact for the specific task at hand. However, we observed a significant higher number of accidental collisions with the physical keyboard for visual representations that do not depict the keyboard in full visual fidelity. In addition, for condition VTInvisible, participants noticed a lower perceived accuracy in aiming for the targets. We actually expected the red bounding box in condition VTOneRow to support users in avoiding accidental collisions, but this is not supported by the experimental results. Hence, further research is needed on appropriate visualizations for non-relevant keyboard areas for a given task at hand.

5.3 Limitations and Future Work

Our work focused on a subset of a possibly large space of input and output mappings. For instance, we didn't explore experiences of augmenting around the keyboard or transform the keyboard when mapping a physical key to an action which we are discussing later in this section. Related, there are additional mappings that can be explored in the future, such as mapping a single key to multiple actions or multiple keys to a coordinate.

Our studies have been based on the physical keyboard Logitech G810 so far, but we see opportunities to explore other types of keyboards including latptop's keyboards. Also, we have been using an external tracker (Optitrack) for our studies, but it would be interesting to see if we could build these type of experiences without relying on fixed trackers in the environment to demonstrate that this approach could really work for mobile users already with today's technology. Also, given the external tracking system, hand and finger actions could be triggered without a keyboard at all. Yet, the act of pressing on a button is subtle and hard to sense by remote sensing, or would require larger gestures. In contrast, using a keyboard button presses can be sensed very accurately and a verification action can be felt. Still, it is valuable to study the effects of using a keyboard vs. on-surface touch for selected tasks in more detail.

In addition, we have results based on the specific experiences we designed. We need to explore more applications to generalize the results. For instance, we studied how to use keys to insert content into Word, but haven't explored yet this type of experience on other types of office productivity applications such as spreadsheet applications. So far, we have used simple visualization in our prototypes, but haven't studied in depth alternate visualizations of physical keyboard reconfiguration in VR.

Moving forward, we are interested in extending the work to combine augmentation of the physical keyboard in VR with augmentation around the keyboard. For instance, window management could be done by displaying all the open applications around the keyboard to allow rapid switching between them. When an application has been selected, the keys of the physical keyboards could be automatically modified accordingly to accommodate the active application. In addition, we are interested in studying if our approach is also working for other type of keyboards. For instance, using a laptop keyboard can be very interesting to explore because we potentially see many people travel with only their laptop and a mobile HMD, a view that is also shared by others [20, 29]. Laptop keyboards have different form factor and key design which is likely to open up possibilities for additional design exploration, for instance by allowing gestures by swiping across the physical keys [83]. To this end, alternative sensing capabilities (such as touch-enabled physical keyboards [61]) could be employed with our use cases. Further, this work focused on the use of individual fingers when operating the keyboard. Future work should also investigate the use of multi-finger input.

Our work demonstrates the rich design possibilities that open up when reconfiguring physical keyboards for VR. However, this idea can be brought even further by exploring using the mouse in VR along with the keyboard. In fact, the mouse could also be augmented based on the context and the task of the user. We could also consider using a touch mouse, which again will open up additional input mapping possibilities. Related, touchpads, which are embedded in most laptops, can also be compelling to augment in VR because it is possible to get a very precise coordinate from it and then use this and other input information to dynamically display information on it and modify it's role, such as swapping between acting as a small touchscreen and acting as an indirect pointing device. In some contexts, we could even just display the touchpad and make the keyboard disappear. Also, the proposed applications could be transferred to Augmented Reality and explored further. One technical difference in AR is the view on the user's physical hands. If the view of the physical hands should be adopted (e.g., for showing minimalistic finger representations as in this paper), there is a need to generate a mask of the hands that enables their display or hiding. One option for video see-through AR systems could be to use chroma keying. Another option would be to render a virtual keyboard (and hands) on top of the physical keyboard.

Additional avenues for future work resides in probing the empirical user experience aspects of this work deeper by further experimentation. For example, our work raises questions about how to best design for perceived affordance, or how to best dynamically reconfigure a keyboard to assist users in complex workflows.

6 Conclusions

Physical keyboards are common peripherals for personal computers and are efficient standard text entry devices. While recent research has investigated how physical keyboards can be used in immersive HMD-based VR, so far, the physical layout of the keyboards has typically been directly transplanted into VR with the explicit goal of replicating typing experiences in a standard desktop environment.

In this paper, we have explored how to fully leverage the immersiveness of VR to change the input and output characteristics of physical keyboard interaction within a VR environment. This allowed us to reconfigure the input and output mappings of both individual keys and the keyboard as a whole. We explored a set of input and output mappings for reconfiguring the virtual presentation of physical keyboards and designed, implemented and evaluated nine VR-relevant applications: emojis, languages and special characters, application shortcuts, virtual text processing macros, window manager, photo browser, a game (whack-a-mole), secure password entry and a virtual touch bar. We investigated the feasibility of the applications in a user study with 20 participants and found that the applications were usable in VR.

From our results we see that we can integrate physical keyboards in VR experiences in many flexible ways. The biggest advantage of standard physical keyboards is that they are actually available as virtually every PC and laptop is already equipped with a physical keyboard. Instead of asking users to remove keyboards during VR use to make space for dedicated VR input devices, it might make more sense to flexibly integrate the keyboard into the VR experience, at least for scenarios such as office work [29]. Keyboards provide haptic feedback that can be used in many ways, for example for virtual keys, sliders, or to simulate reactive surfaces such as the case of whack-a-mole. They also provide accurate tactile guidance for the users' fingers. For many input tasks, they are the fastest and most accurate input device available.

In conclusion, we have shown that physical keyboards can be used very flexibly as an input device for many different tasks in VR and could instantaneously reconfigure based on the context. We believe that their unique advantages will make physical keyboards promising and flexible input devices for many VR experiences in the future.

References

[1] F. A. Alsulaiman and A. El Saddik. A novel 3d graphical password schema. In *2006 IEEE Symposium on Virtual Environments, Human-Computer Interfaces and Measurement Systems*, pp. 125–128. IEEE, 2006.

[2] Apple. Apple touch bar. `https://developer.apple.com/macos/touch-bar/`. Last accessed 19.03.2019.

[3] M. Azmandian, M. Hancock, H. Benko, E. Ofek, and A. D. Wilson. Haptic retargeting: Dynamic repurposing of passive haptics for enhanced virtual reality experiences. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pp. 1968–1979. ACM, New York, NY, USA, 2016. doi: 10.1145/2858036.2858226

[4] G. Bailly, T. Pietrzak, J. Deber, and D. J. Wigdor. Métamorphe: augmenting hotkey usage with actuated keys. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 563–572. ACM, 2013.

[5] F. Block, H. Gellersen, and N. Villar. Touch-display keyboards: transforming keyboards into interactive surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1145–1154. ACM, 2010.

[6] S. Bovet, A. Kehoe, K. Crowley, N. Curran, M. Gutierrez, M. Meisser, D. O. Sullivan, and T. Rouvinez. Using traditional keyboards in vr: Steamvr developer kit and pilot game user study. In *2018 IEEE Games, Entertainment, Media Conference (GEM)*, pp. 1–9. IEEE, 2018.

[7] J. Brooke et al. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[8] E. Burns, S. Razzaque, M. Whitton, and F. Brooks. Macbeth: The avatar which i see before me and its movement toward my hand. in proceedings of the 2016 chi conference on human factors in computing systems. In *Proceedings of IEEE Virtual Reality Conference*, p. 295–296. IEEE, 2007.

[9] D. Buschek, B. Roppelt, and F. Alt. Extending keyboard shortcuts with arm and wrist rotation gestures. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, p. 21. ACM, 2018.

[10] S. K. Card, T. P. Moran, and A. Newell. *The psychology of human-computer interaction*. Lawrence Erlbaum Associates, 1983.

[11] X. Chen, T. Grossman, D. J. Wigdor, and G. Fitzmaurice. Duet: exploring joint interactions on a smart phone and a smart watch. In

Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 159–168. ACM, 2014.

[12] O. E. H. C. B. H. Cheng, Lung-Pan and A. D Wilson. Sparse haptic proxy: Touch feedback in virtual environments using a general passive prop. in proceedings of the 2017 chi conference on human factors in computing systems. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, p. 3718–3728. ACM, 2017.

[13] F. Conti and O. Khatib. Spanning large workspaces using small haptic devices. in proceedings of first joint eurohaptics conference and symposium on haptic interfaces for virtual environment and teleoperator. In *roceedings of First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator*, p. 183–188. ACM, 2005.

[14] C. Corsten, I. Avellino, M. Möllers, and J. Borchers. Instant user interfaces: repurposing everyday objects as input devices. in proceedings of the acm international conference on interactive tabletops and surfaces (its '13). In *Proceedings of the ACM international conference on Interactive tabletops and surfaces*, pp. 71–80. ACM, 2013.

[15] J. A. de Guzman, K. Thilakarathna, and A. Seneviratne. Security and privacy approaches in mixed reality: A literature survey. *arXiv preprint arXiv:1802.05797*, 2018.

[16] V. Dhakal, A. M. Feit, P. O. Kristensson, and A. Oulasvirta. Observations on typing from 136 million keystrokes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pp. 646:1–646:12. ACM, New York, NY, USA, 2018. doi: 10.1145/3173574.3174220

[17] P. H. Dietz, B. Eidelson, J. Westhues, and S. Bathiche. A practical pressure sensitive computer keyboard. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pp. 55–58. ACM, 2009.

[18] T. Dube and A. Arif. Text entry in virtual reality: A comprehensive review of the literature. In *Proceedings of HCI International 2019*, 2019.

[19] M. Eiband, M. Khamis, E. Von Zezschwitz, H. Hussmann, and F. Alt. Understanding shoulder surfing in the wild: Stories from users and observers. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 4254–4265. ACM, 2017.

[20] Engadget. Flying with a vr headset isn't as dorky as it sounds. `https://www.engadget.com/2018/02/22/htc-vive-focus-in-flight-vr/`. Last accessed 19.03.2019.

[21] W. Fallot-Burghardt, M. Fjeld, C. Speirs, S. Ziegenspeck, H. Krueger, and T. Läubli. Touch&type: a novel pointing device for notebook computers. In *Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles*, pp. 465–468. ACM, 2006.

[22] G. W. Fitzmaurice, H. Ishii, and A. S. W. Buxton. Bricks: laying the foundations for graspable user interfaces. in proceedings of the sigchi conference on human factors in computing systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 442–449. ACM, 1995.

[23] M.-C. Fluet, O. Lambercy, and R. Gassert. Effects of 2d/3d visual feedback and visuomotor collocation on motor performance in a virtual peg insertion test. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 4776–4779. IEEE, 2012.

[24] S. Follmer, D. Leithinger, A. Olwal, A. Hogge, and H. Ishii. inform: dynamic physical affordances and constraints through shape and object actuation. in proceedings of the acm user interface software and technology symposium. In *Proceedings of the ACM User Interface Software and Technology Symposium*, p. 417–426. ACM, 2013.

[25] M. J. Fu, A. D. Hershberger, K. Sano, and M. C. Çavuşoğlu. Effect of visuomotor colocation on 3d fitts' task performance in physical and virtual environments. *Presence*, 21(3):305–320, 2012.

[26] H. Gellersen and F. Block. Novel interactions on the keyboard. *Computer*, 45(4):36–40, 2012.

[27] C. George, M. Khamis, E. von Zezschwitz, M. Burger, H. Schmidt, F. Alt, and H. Hussmann. Seamless and secure vr: Adapting and evaluating established authentication systems for virtual reality. NDSS, 2017.

[28] K. R. Gray. Facilitating keyboard use while wearing a head-mounted display. 2018.

[29] J. Grubert, E. Ofek, M. Pahud, and P. O. Kristensson. The office of the future: Virtual, portable, and global. *IEEE computer graphics and applications*, 38(6):125–133, 2018.

[30] J. Grubert, L. Witzani, E. Ofek, M. Pahud, M. Kranz, and P. O. Kristensson. Effects of hand representations for typing in virtual reality. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 151–158, March 2018. doi: 10.1109/VR.2018.8446250

[31] J. Grubert, L. Witzani, E. Ofek, M. Pahud, M. Kranz, and P. O. Kristensson. Text entry in immersive head-mounted display-based virtual reality using standard keyboards. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 159–166, March 2018. doi: 10.1109/VR.2018.8446059

[32] I. Habib, N. Berggren, E. Rehn, G. Josefsson, A. Kunz, and M. Fjeld. Dgts: Integrated typing and pointing. In *IFIP Conference on Human-Computer Interaction*, pp. 232–235. Springer, 2009.

[33] A. Hettiarachchi and D. Wigdor. Annexing reality: Enabling opportunistic use of everyday objects as tangible proxies in augmented reality. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, p. 1957–1967. ACM, 2016.

[34] K. Hinckley, R. Pausch, J. C. Goble, and N. F. Kassell. Passive real-world interface props for neurosurgical visualization. in proceedings of the sigchi conference on human factors in computing systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 452–458. ACM, 1994.

[35] A. H. Hoppe, L. Otto, F. van de Camp, R. Stiefelhagen, and G. Unmüßig. qvrty: Virtual keyboard with a haptic, real-world representation. In *International Conference on Human-Computer Interaction*, pp. 266–272. Springer, 2018.

[36] C. Hughes, C. Stapleton, D. Hughes, and E. Smith. Mixed reality in education, entertainment, and training. *Computer Graphics and Applications*, 25(6):24–30, 2005.

[37] I. U. T. Inc. A brief history of the lcd key technology. `http://www.lcd-keys.com/english/history.htm`. Last accessed 19.03.2019.

[38] J. Kato, D. Sakamoto, and T. Igarashi. Surfboard: keyboard with microphone as a low-cost interactive surface. In *Adjunct proceedings of the 23nd annual ACM symposium on User interface software and technology*, pp. 387–388. ACM, 2010.

[39] D. Kim, S. Izadi, J. Dostal, C. Rhemann, C. Keskin, C. Zach, J. Shotton, T. Large, S. Bathiche, M. Nießner, et al. Retrodepth: 3d silhouette sensing for high-precision input on and above physical surfaces. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pp. 1377–1386. ACM, 2014.

[40] P. Knierim, V. Schwind, A. M. Feit, F. Nieuwenhuizen, and N. Henze. Physical keyboards in virtual reality: Analysis of typing performance and effects of avatar hands. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, p. 345. ACM, 2018.

[41] D. A. Kontarinis, J. S. Son, W. Peine, and R. D. Howe. A tactile shape sensing and display system for teleoperated manipulation. in proceedings of the ieee conference on robotics and automation. In *Proceedings of the IEEE Conference on Robotics and Automation*, p. 641–646. IEEE, 1995.

[42] K. J. Kruszy´nski and R. van Liere. Tangible props for scientific visualization: concept, requirements. *Virtual reality*, 13(4):235–244, 2009.

[43] T. Kurosawa, B. Shizuki, and J. Tanaka. Keyboard clawing: input method by clawing key tops. In *International Conference on Human-Computer Interaction*, pp. 272–280. Springer, 2013.

[44] D. M. Lane, H. A. Napier, S. C. Peres, and A. Sandor. Hidden costs of graphical user interfaces: Failure to make the transition from menus and icon toolbars to keyboard shortcuts. *International Journal of Human-Computer Interaction*, 18(2):133–144, 2005.

[45] K. Lebeck, K. Ruth, T. Kohno, and F. Roesner. Securing augmented reality output. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 320–337. IEEE, 2017.

[46] B. Lee, H. Park, and H. Bang. Multidirectional pointing input using a hardware keyboard. *ETRI Journal*, 35(6):1160–1163, 2013.

[47] J. R. Lewis. Psychometric evaluation of an after-scenario questionnaire for computer usability studies: the asq. *ACM Sigchi Bulletin*, 23(1):78–81, 1991.

[48] J.-W. Lin, P.-H. Han, J.-Y. Lee, Y.-S. Chen, T.-W. Chang, K.-W. Chen,

and Y.-P. Hung. Visualizing the keyboard in virtual reality for enhancing immersive experience. In *ACM SIGGRAPH 2017 Posters*, p. 35. ACM, 2017.

[49] R. W. Lindeman, J. L. Sibert, and J. K. Hahn. Hand-held windows: towards effective 2d interaction in immersive virtual environments. In *Proceedings IEEE Virtual Reality (Cat. No. 99CB36316)*, pp. 205–212. IEEE, 1999.

[50] Logitech. Logitech g19 keyboard for gaming. `https://support.logitech.com/en_us/product/g19-keyboard-for-gaming`. Last accessed 19.03.2019.

[51] K.-L. Low, G. Welch, A. Lastra, and H. Fuchs. Life-sized projector-based dioramas. in proceedings of the acm symposium on virtual reality software and technology (vrst '01). In *Proceedings of the IEEE Conference on Robotics and Automation*, pp. 93–101. ACM, 2001.

[52] C. C. Loy, W. Lai, and C. Lim. Development of a pressure-based typing biometrics user authentication system. *ASEAN Virtual Instrumentation Applications Contest Submission*, 2005.

[53] A. Maiti, M. Jadliwala, and C. Weber. Preventing shoulder surfing using randomized augmented reality keyboards. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pp. 630–635. IEEE, 2017.

[54] N. Marquardt, R. Jota, S. Greenberg, and J. A. Jorge. The continuous interaction space: interaction techniques unifying touch and gesture on and above a digital surface. In *IFIP Conference on Human-Computer Interaction*, pp. 461–476. Springer, 2011.

[55] M. McGill, D. Boland, R. Murray-Smith, and S. Brewster. A dose of reality: overcoming usability challenges in vr head-mounted displays. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 2143–2152. ACM, 2015. doi: 10.1145/2702123.2702382

[56] P. Milgram and F. Kishino. A taxonomy of mixed reality visual displays. *IEICE Transactions on Information and Systems*, 77(12):1321–1329, 1994.

[57] D. Norman. *The design of everyday things: Revised and expanded edition*. Basic books, 2013.

[58] P. K. Novak, J. Smailović, B. Sluban, and I. Mozetič. Sentiment of emojis. *PloS one*, 10(12):e0144296, 2015.

[59] R. C. Omanson, C. S. Miller, E. Young, and D. Schwantes. Comparison of mouse and keyboard efficiency. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 54, pp. 600–604. Sage Publications Sage CA: Los Angeles, CA, 2010.

[60] optimus. Optimus oled keyboards. `https://www.artlebedev.com/optimus/`. Last accessed 19.03.2019.

[61] A. Otte, D. Schneider, T. Menzner, T. Gesslein, P. Gagel, and J. Grubert. Evaluating text entry in virtual reality using a touch-sensitive physical keyboard. In *2019 IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 2019.

[62] J. Ramos, Z. Li, J. Rosas, N. Banovic, J. Mankoff, and A. Dey. Keyboard surface interaction: Making the keyboard into a pointing device. *arXiv preprint arXiv:1601.04029*, 2016.

[63] Razor. Razor deathstalker ultimate keyboard. `https://support.razer.com/gaming-keyboards/razer-deathstalker-ultimate`. Last accessed 19.03.2019.

[64] J. Rekimoto, T. Ishizawa, C. Schwesig, and H. Oba. Presense: interaction techniques for finger sensing input devices. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, pp. 203–212. ACM, 2003.

[65] F. Roesner, T. Kohno, and D. Molnar. Security and privacy for augmented reality systems. *Commun. ACM*, 57(4):88–96, 2014.

[66] A. Roudaut, D. Krusteva, M. McCoy, A. Karnik, K. Ramani, and S. Subramanian. Cubimorph: designing modular interactive devices. in robotics and automation (icra). In *Proceedings of the IEEE Conference on Robotics and Automation*, p. 3339–3334. IEEE, 2016.

[67] A. Roudaut, R. Reed, T. Hao, and S. Subramanian. changibles: analyzing and designing shape changing constructive assembly. in proceedings of the 32nd annual acm conference on human factors in computing systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, p. 2593–2596. ACM, 2014.

[68] M. A. Sasse and I. Flechais. Usable security: Why do we need it? how do we get it? O'Reilly, 2005.

[69] K. Sekimori, Y. Yamasaki, Y. Takagi, K. Murata, B. Shizuki, and S. Takahashi. Ex-space: Expanded space key by sliding thumb on home position. In *International Conference on Human-Computer Interaction*, pp. 68–78. Springer, 2018.

[70] Y. Shi, T. Vega Gálvez, H. Zhang, and S. Nanayakkara. Gestakey: Get more done with just-a-key on a keyboard. In *Adjunct Publication of the 30th Annual ACM Symposium on User Interface Software and Technology*, pp. 73–75. ACM, 2017.

[71] Y. Shi, H. Zhang, H. Rajapakse, N. T. Perera, T. Vega Gálvez, and S. Nanayakkara. Gestakey: Touch interaction on individual keycaps. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, p. 596. ACM, 2018.

[72] A. Strauss and J. Corbin. *Basics of qualitative research*. Sage publications, 1990.

[73] S. Taylor, C. Keskin, O. Hilliges, S. Izadi, and J. Helmes. Type-hover-swipe in 96 bytes: A motion sensing mechanical keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pp. 1695–1704. ACM, New York, NY, USA, 2014. doi: 10.1145/2556288.2557030

[74] S. Taylor, C. Keskin, O. Hilliges, S. Izadi, and J. Helmes. Type-hover-swipe in 96 bytes: a motion sensing mechanical keyboard. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pp. 1695–1704. ACM, 2014.

[75] R. J. Teather, R. S. Allison, and W. Stuerzlinger. Evaluating visual/motor co-location in fish-tank virtual reality. In *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*, pp. 624–629. IEEE, 2009.

[76] Y.-C. Tung, T. Y. Cheng, N.-H. Yu, C. Wang, and M. Y. Chen. Flickboard: Enabling trackpad interaction with automatic mode switching on a capacitive-sensing keyboard. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 1847–1850. ACM, 2015.

[77] VIVE. Introducing the logitech bridge sdk. `https://blog.vive.com/us/2017/11/02/introducing-the-logitech-bridge-sdk/`. Last accessed 19.03.2019.

[78] J. Walker, S. Kuhl, and K. Vertanen. Decoder-assisted typing using an HMD and a physical keyboard. In *CHI 2016 Workshop on Inviscid Text Entry and Beyond*, p. unpublished, 2016.

[79] J. Walker, B. Li, K. Vertanen, and S. Kuhl. Efficient typing on a visually occluded physical keyboard. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 5457–5461. ACM, 2017.

[80] A. D. Wilson. Robust computer vision-based detection of pinching for one and two-handed gesture input. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, pp. 255–258. ACM, 2006.

[81] W. L. Zagler, C. Beck, and G. Seisenbacher. *FASTY-faster and easier text generation for disabled people*. na, 2003.

[82] H. Zhang and Y. Li. Gestkeyboard: enabling gesture-based interaction on ordinary physical keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1675–1684. ACM, 2014.

[83] H. Zhang and Y. Li. Gestkeyboard: Enabling gesture-based interaction on ordinary physical keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pp. 1675–1684. ACM, New York, NY, USA, 2014. doi: 10.1145/2556288.2557362

[84] Y. Zhao and S. Follmer. A functional optimization based approach for continuous 3d retargeted touch of arbitrary, complex boundaries in haptic virtual reality. in proceedings of the 2018 chi conference on human factors in computing systems. In *Proceedings of the 2018 CHI Conference on Computing Systems*. ACM, 2018.

[85] J. Zheng, B. Lewis, J. Avery, and D. Vogel. Fingerarc and fingerchord: Supporting novice to expert transitions with guided finger-aware shortcuts. In *The 31st Annual ACM Symposium on User Interface Software and Technology*, pp. 347–363. ACM, 2018.

[86] J. Zheng and D. Vogel. Finger-aware shortcuts. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 4274–4285. ACM, 2016.