

Extending Movable Surfaces with Touch Interaction Using the VirtualTablet: An Extended View

Adrian H. Hoppe^{*1}, Felix Marek¹, Florian van de Camp², Rainer Stiefelwagen¹

¹Karlsruhe Institute of Technology (KIT), Institute for Anthropomatics and Robotics (IAR), cv:hci Lab, 76131 Karlsruhe, Germany

²Fraunhofer IOSB, Interactive Analysis and Diagnosis (IAD), 76131 Karlsruhe, Germany

ARTICLE INFO

Article history:

Received: 07 January, 2020

Accepted: 04 February, 2020

Online: 26 March, 2020

Keywords:

VirtualTablet

Virtual environment

Virtual reality

Touch interaction

Haptic feedback

ABSTRACT

Immersive output and natural input are two core aspects of a virtual reality experience. Current systems are frequently operated by a controller or gesture-based approach. However, these techniques are either very accurate but require an effort to learn, or very natural but miss haptic feedback for optimal precision. We transfer ubiquitous touch interaction with haptic feedback into a virtual environment. To validate the performance of our implementation, we performed a user study with 28 participants. As the results show, the movable and cheap real world object supplies an accurate touch detection that is equal to a laserpointer-based interaction with a controller. Moreover, the virtual tablet can extend the functionality of a real world tablet. Additional information can be displayed in mid-air around the touchable area and the tablet can be turned over to interact with both sides. Therefore, touch interaction in virtual environments allows easy to learn and precise system interaction and can even augment the established touch metaphor with new paradigms.

1 Introduction

Virtual Reality (VR) has established itself at a consumer level. Many different VR systems immerse the user in an interactive environment. However, the different systems all have their distinct input devices. For newer users, we witnessed, it is quite difficult to just *pull the trigger* or *press the grip button*, since users do not see their own hands, but only the floating input device. More natural interfaces are needed. Gesture interaction via a camera that is mounted on the Head Mounted Display (HMD) lets the user grab virtual objects and manipulate the Virtual Environment (VE) effortlessly. But, haptic sensations are missing.

We therefore present VirtualTablet - A touchable object that is made of very cheap materials and can have any size or shape. This paper is an extension of work originally presented in the IEEE Conference on Virtual Reality and 3D User Interfaces (VR) [1]. It presents further detail regarding related work, implementation and design as well as evaluation and results. This paper makes two technical contributions: First, VirtualTablet supplies touch interaction on a movable surface captured by a movable camera. Second, different

interaction techniques are presented that increase the functionality of the tablet to more than the capability of a real-world tablet.

2 Related Work

Haptic feedback increases performance and usability [2]–[4]. Different systems can be used to realize haptic feedback for a user. With active haptics, users wear a device that exerts pressure or a force on the user's hand. Scheggi et al. [5] use small tactile interfaces at the fingertips. The interfaces press down on the fingers if an object is touched. The VRGluV¹ is an exoskeleton that allows to constrain the movement of the fingers, thereby providing haptic feedback. With passive haptics the user touches real objects to receive feedback. The real objects can be tracked, for example with markers, and then displayed in the VE. Simeone et al. [6] track different real objects, e.g. a cup or an umbrella, as well as the head and hands of the user. They showed that the virtually displayed object does not need to match the real object perfectly. Some variation of material properties or shape allow the use of a broader range of virtual objects even if the set of real world objects is limited.

*Adrian H. Hoppe, Fraunhoferstr. 1, 76131 Karlsruhe, Germany, +49 721 6091 307 & adrian.hoppe@kit.edu

¹see <https://vrgluv.com/>

If users start navigating in the VE, a mismatch between the location of the real and virtual objects may occur. To overcome this, movable haptic proxies or redirection techniques can be used. Araujo et al. [7] use a robotic arm to move a haptic surface to block the user's hand when it gets close to a virtual object. The head of the robotic arm can be rotated and exchanged, hereby providing a wide range of haptic surfaces. i.e. different textures, a pressure sensor, an interactive surface, physical controls or even a heat emitter. Kohli [8] implemented a touch interaction using an IMPULSE system that tracks the finger via a LED. Utilizing space warping, it is possible to alter the movement of the user's hand in the real world, without the user noticing. By steering the hand towards a real world object, haptic feedback can be given by a virtual object at another position. Azmandian et al. [9] track cubes and a wearable glove with markers on a table surface. The user's hand and forearm, as well as some virtual cubes are displayed in the VE. By redirecting the hand of the user while reaching for the virtual objects, they provide haptic sensations for several virtual cubes with a single real world cube. Cheng et al. [10] apply the same principle. Using eye tracking and an OptiTrack system they predict what virtual object will be touched. The movement of the virtual hand is then modified so that a user corrects their real world movement. This allows Cheng et al. to only use a hemispherical surface as a touch proxy. Most of these techniques require a rather static scene and a non navigating user and are thereby not suitable for full body VR experiences.

The presented systems above all give a haptic feeling to the user, but have limited input capabilities. To overcome this, Medeiros et al. [11] use a tracked tablet in a CAVE environment. Depending on the orientation of the tablet, the respective view into the virtual world is rendered on the touch screen. A user can select and manipulate objects by clicking on them in the rendered view or using a set of control buttons. The tablet is tracked by an external tracking system using markers. Xiao et al. [12] use the internal sensors of a Microsoft HoloLens to allow touch interaction on static surfaces. The system is much more flexible as all of the above systems, since it does not require the user to put on gloves or to set up an external tracking system. Using the depth sensors and a segmentation algorithm the hands of the user are recognized and the fingertips extracted. A RANSAC algorithm detects surfaces in the room. Combining these allows for touch interaction with haptic feedback on a real world surface. There is no need to visualize the hands or the surface for the user, since the HoloLens is an augmented reality (AR) HMD and the user sees the real objects. However, only static surfaces are used.

3 Implementation

The VirtualTablet system is implemented on a HTC Vive Pro VR HMD². The Vive features a 1440×1600 pixels display per eye with a 90 Hz refresh rate. The field of view of the Vive is 110° . Using the Lighthouse tracking system and an IMU the pose of the HMD is detected in a room area of $5 \times 5 m^2$. Moreover, the Vive has a stereo camera on the front of the HMD³. Each camera has a resolution of

²see <https://www.vive.com/de/product/vive-pro/>

³see <https://developer.vive.com/resources/knowledgebase/intro-vive-srworks-sdk/> and <https://www.stereolabs.com/blog/vive-pro-ar-zed-mini/>

640×480 pixels with a refresh rate of 60 Hz and a field of view of 96° horizontal and 80° vertical. The camera can be accessed with the SRWorks toolkit with a latency of about 100 ms with SRWorks to 200 ms with the ZED Mini. SRWorks supplies depth information with an accuracy of ± 3 cm at 1 m and ± 10 cm at 2 m distance from the camera. The minimal distance is 30 cm.

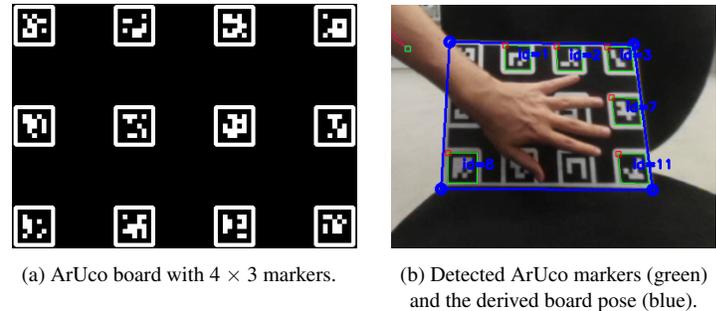


Figure 1: Setup and Detection of the ArUco board.

To detect the movable objects, ArUco markers [13] are used. They provide a position and rotation in respect to the camera, given their real world size. The VirtualTablet consists of a simple rigid base, for example a piece of acrylic glass or cardboard, pasted up with ArUco markers. Because the hand of the user will overlap with some markers while interacting, the markers are spread out over the board (see Figure 1). ArUco markers can be uniquely identified, therefore allowing the use of more than one VirtualTablet at the same time.

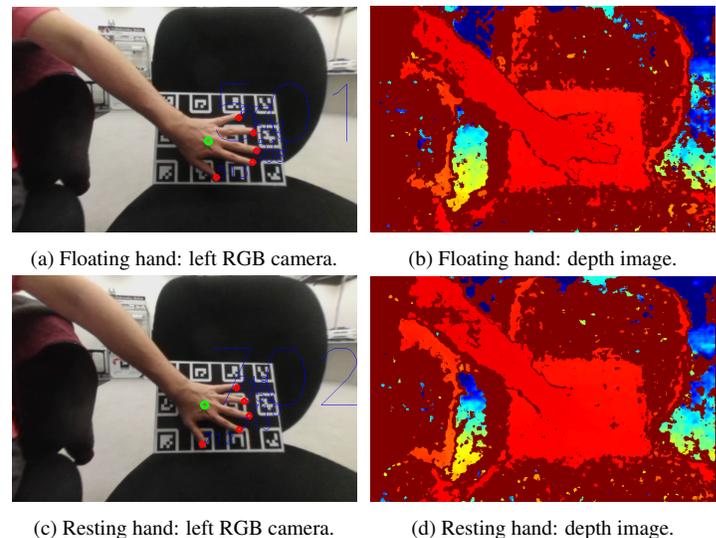


Figure 2: Front camera images of a hand hovering above the board and touching the board. Dark red color in the depth image mark invalid depth information.

To detect the fingertips, a segmentation algorithm is used, since the hand of the user cannot be reconstructed from the given depth information (see Figure 2). First, the hand is extracted from the RGB image by first masking out the board. An interacting hand will be inside this area. Since the ArUco board only consists of black and white colors, we subtract these colors from the image which results in a binary representation of the segmented hand. Pixels

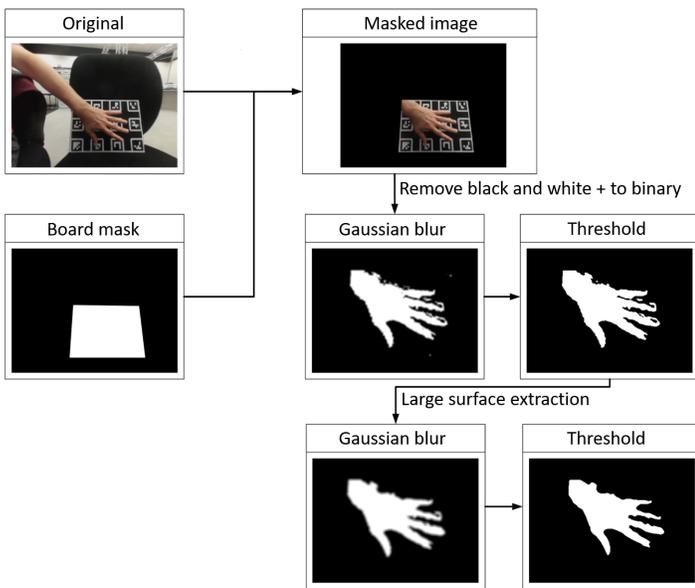


Figure 3: Hand segmentation process.

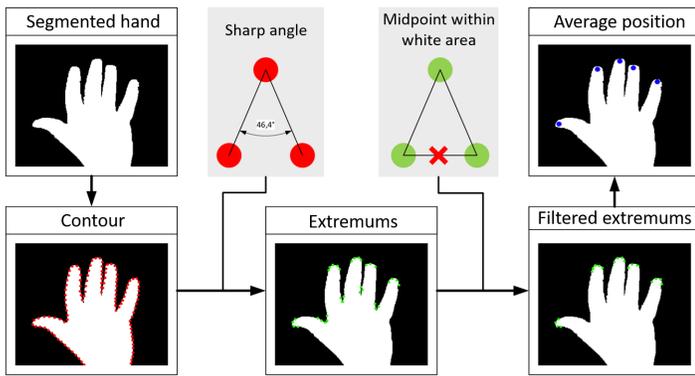


Figure 4: Fingertip detection process.

with a black or white color are removed in the HSV color space at a threshold with a low saturation (white) or a low brightness (black), i.e. $S \leq 80$ or $V \leq 40$ (scale from 0 to 255). As Figure 3 shows, the binary image is then enhanced by a blur filter, the application of a threshold and the extraction of larger surfaces⁴. There are other ways to extract the hand by using skin color, so this approach can be easily exchanged if the background colors or lighting conditions change. Second, the fingertips are detected by calculating the extreme values of the contour of the hand [14]. This is done by calculating the angle between three points on the contour that are for our camera setup 18 pixels apart from each other (see Figure 4). To track the fingertips from one frame to another a distance-based tracking algorithm similar to [12] is used. Combined with the depth information from the stereo camera the 2D tracked location of the user's fingers can be projected into the 3D space relative to the HMD. To have a more robust depth value an average of 11×11 pixels is calculated around the 2D fingertip location. Since the pose of the surface and the fingertips are known, a touch can be detected. If the finger is closer than 0.5 cm to the surface, a click is triggered.

⁴see <https://medium.com/@muehler.v/simple-hand-gesture-recognition-using-opencv-and-javascript-eb3d6ced28a0>

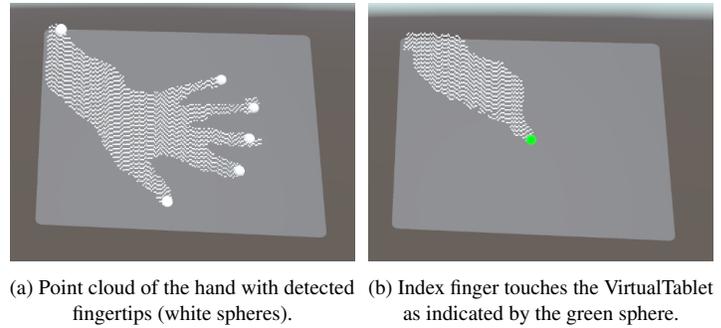


Figure 5: Visualization of the VirtualTablet and the user's hand.

To visualize the VirtualTablet and the user's hands in VR, the engine Unity3D⁵ is used. At the location of the real tablet, a virtual tablet with the same size and shape is rendered (see Figure 5). To give the user a visual feedback of the location of her or his hand, a 3D point cloud is rendered. White spheres at the fingertips highlight currently tracked fingers. The sphere turns green if a touch is detected.

The VirtualTablet can be used as normal tablet which is familiar to many users. However, VR allows to enhance and extend the interaction with the tablet. The following subsections describe different techniques that are enabled by VirtualTablet.

3.1 Moving the VirtualTablet

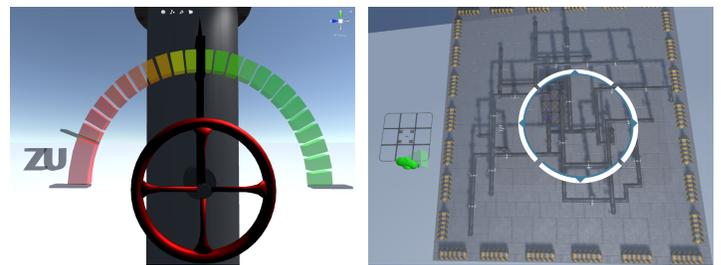


Figure 6: Examples for a rotating tablet interaction.

Even without the touch recognition the VirtualTablet allows different forms of interaction. The position and rotation of the tablet can be used as an input to e.g. open a menu or change a value if held inside a trigger volume. If a user holds the tablet into a virtual object, an action can be triggered. By rotating the tablet at the location of a wheel, the wheel can be turned (see Figure 6a). We implemented a navigation mechanism that allows the user to move a map around by tilting the tablet (see Figure 6b). If the user touches a location she or he is teleported there.

Since the VirtualTablet has ArUco markers on both sides, it can recognize touch on both surfaces. This can be used to give the

⁵see <https://unity3d.com/>

user a more natural way to navigate through menus. If the user flips the tablet around, a different user interface is displayed. In our application it is used to switch between an information interface and a map display for navigation.

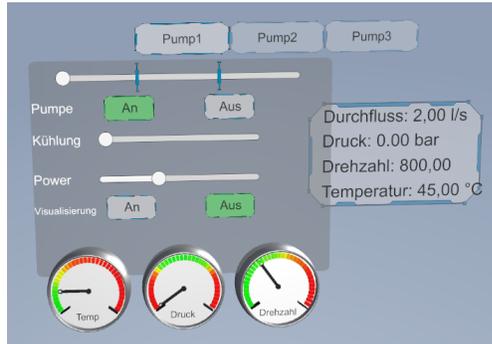


Figure 7: A menu that extends non-interactable information over the edges of the physical tablet.

3.2 Extending the VirtualTablet

VR allows to display information at any given location. In reality, displays are much more constrained and information cannot (yet) be displayed in mid-air. The VirtualTablet allows to extend the displayed content over the edges of the physical tablet (see Figure 7). All interactable elements, like buttons and sliders, are displayed on the touchable surface. Other informations are arranged around the tablet.

3.3 Duplicating the VirtualTablet



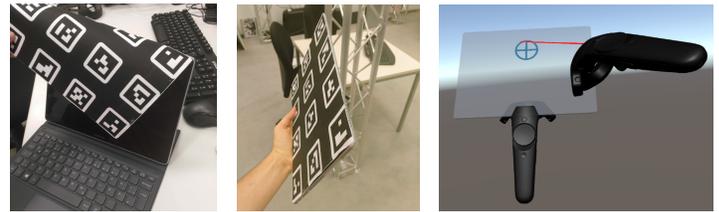
(a) Mobile acrylic glass.

(b) Mounted cardboard.

Figure 8: Different interaction surfaces.

Physical tablets are expensive, rely on a power supply and only come in distinct shapes and sizes. With our approach multiple

interactable surfaces can be created fast and cheap. As seen in Figure 8 we created a small mobile tablet and a larger mounted interaction surface. The different surfaces can be used in parallel.



(a) Attaching the ArUco board to the capacitive touchscreen tablet.

(b) Holding the capacitive touchscreen tablet.

(c) Laserpointer and interaction surface attached to the HTC Vive controllers.

Figure 9: Setup of the input techniques for comparison with the VirtualTablet.

4 Evaluation

To evaluate the proposed system and interaction methods, a user study was performed. We designed an application that allows a user to control different machines and valves in a factory using the VirtualTablet. The task of the user was to react to a breach in the pipe system. While solving the issues, the user interacted repeatedly with the extended interaction methods described above.

4.1 Independent Variables

To compare the detection performance of the VirtualTablet, other input methods were implemented. A capacitive touchscreen tablet was equipped with a ArUco board (see Figure 9a and 9b). The capacitive tablet uses the same hand visualization as the VirtualTablet, but the touch detection from the display as a ground truth. The size of the tablet is about 1 cm smaller in width and 0.5 cm smaller in height than the acrylic glass VirtualTablet. The default interaction tools for the HTC Vive are the provided controllers. The pose of the controllers is tracked with sub-millimeter accuracy [15]. Applications often use a laser pointer to interact with a handheld menu (see Figure 9c). This virtual pointer technique is effective and efficient [16]. Furthermore, a normal capacitive tablet in a non-VR scenario was used to evaluate the ground truth precision of touch interaction for the given tasks.

The user study was performed in a within-subjects design. The independent variables are the the four described techniques VirtualTablet, CapacitiveTablet and Controller (all in VR) as well as the NonVRTablet. To compensate for the effects of learning and fatigue the conditions were counterbalanced.

4.2 Procedure

First, the users tested the extended interaction methods in the factory application. Second, the users were asked to perform several click and draw interactions with the tablet as in [12]. Users were presented with 6 targets which appeared four times each per technique. Each target had a size of 5×5 cm, independent of the condition. As a result each user performed $6 \times 4 \times 4 = 96$ clicks in total. The order of the click targets was random, but pre-calculated for each

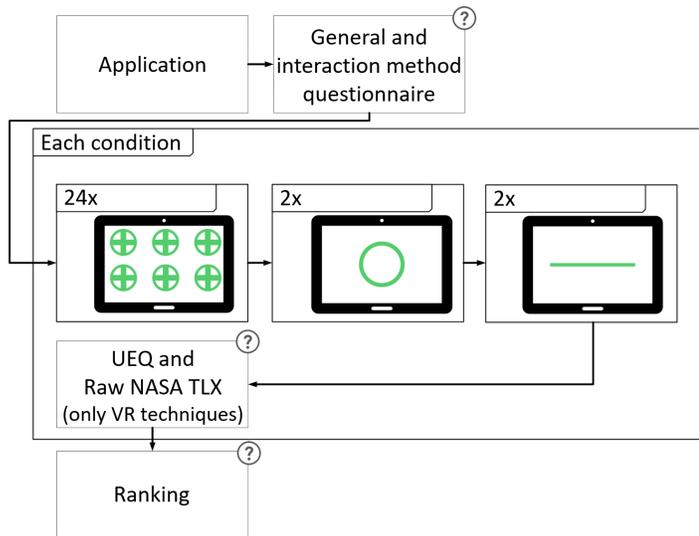


Figure 10: Procedure of the user study. Question mark indicates questionnaire.

technique so each user had the same sequence. For the shape tracing task, users were asked to trace a given figure (circle and line) as closely as possible and finishing their drawing by clicking a button. Each shape was drawn twice. The full procedure is listed in Figure 10.

4.3 Participants

28 people (4 female) participated in the user study with an age distribution of 2 under 20 years, 13 from 21-30 years, 8 from 31-40 years and 5 larger than 40 years. All of which had prior experience with touch input. On a scale of 1 (none) to 7 (very much), users had $\bar{O} 3.2 \pm 1.9$ experience with VR, 15 users had used a tracked controller before.

5 Analysis

5.1 Selection

The precision of the target selection was measured by recording the position of the touch and calculating the distance to the center of the target. We removed 49 outlier points (1.8%) with a distance of more than three standard deviations from the target point (as in [12], [17]). Also, the time taken was measured.

5.2 Tracing

For the shape tracing tasks, we measured how much of the target area was filled. Furthermore, the percentage of pixels painted inside the image was calculated. This value is an indication how accurately a user could draw inside the target area. Moreover, we calculated the average distance of the touch point to the target shape with the width of the brush and the shape removed. Also the duration of drawing was measured.

5.3 Performance and User Experience

To assess the qualitative performance and experience of the different hardware setups, we used Raw NASA TLX [18] and the shortened version of UEQ, the UEQ-S [19] as a usability measurement. These questionnaires were only collected for the VR techniques since the NonVRTablet is only used as a ground truth for precision. At last, users were asked to rank the three VR techniques on a 7-point Likert scale regarding wearing comfort, quality of input and an overall ranking.

5.4 Statistical analysis

We present the results as average value (\bar{O}) with standard deviation (\pm). We used the *rTOST* [20] as a equivalence test and the *Mann-Whitney-Test* [21] as a significance test. Both tests do not assume a normal distribution of the sample set. To calculate the power of the effects, *Cohen's d* [22] is used.

6 Results

6.1 Selection accuracy

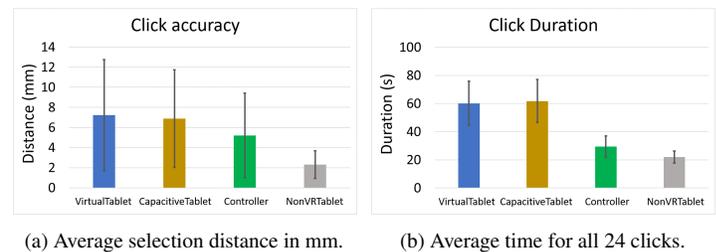


Figure 11: Accuracy and duration of selections.

Table 1: Average selection distance in mm. The 6 targets are coded as T/B = top or bottom row and L/C/R = left, center or right column.

	Total	TL	TC	TR	BL	BC	BR
VirtualTablet							
\bar{O}	6,877	6,528	5,401	7,155	8,005	7,272	6,910
\pm	4,869	4,701	3,603	4,827	5,287	5,287	4,985
CapacitiveTablet							
\bar{O}	6,488	6,553	6,491	7,218	6,752	5,877	6,023
\pm	4,077	3,765	4,198	3,853	4,790	4,093	3,581
Controller							
\bar{O}	4,827	5,028	5,039	5,221	4,905	4,162	4,614
\pm	3,291	3,445	2,929	3,619	3,345	2,783	3,501
NonVRTablet							
\bar{O}	2,246	2,527	2,118	2,301	2,562	2,159	1,820
\pm	1,231	1,300	1,160	1,287	1,308	1,033	1,148

The results of the selection tasks are listed in Figure 11a and Table 1. The three VR input methods VirtualTablet, CapacitiveTablet

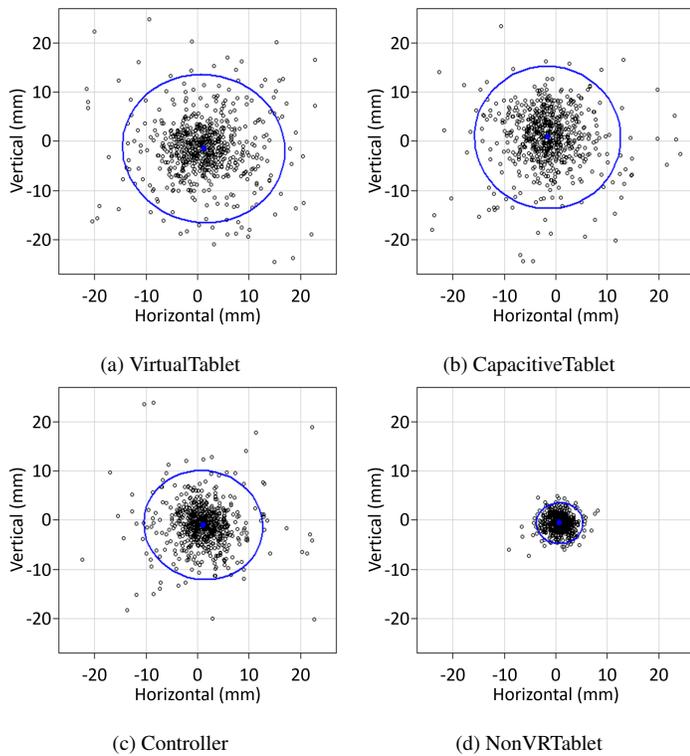


Figure 12: 95% confidence ellipses.

Table 2: rTOST equivalence test between pairs of VirtualTablet (VT), CapacitiveTablet (CT) and Controller (C)

	VT - C	VT - CT	CT - C
Epsilon	2,315	2,315	2,315
Avg. diff.	1,652	-0,001	-1,658
df	687,239	759,937	778,250
Cohen's d	0,412	0,065	0,370
p value	0,003	0,000	0,001

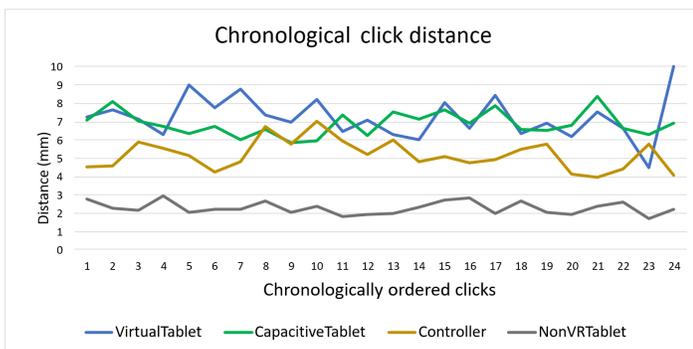


Figure 13: Average distance of clicks chronologically ordered.

and Controller have a selection accuracy of about 5 to 7 mm. The 95% confidence ellipses in Figure 12 show visually, what the rTOST test (see Table 2) confirms. All three VR input methods are equivalent. The epsilon of 2.315 is the average distance of the ground truth tablet interaction, meaning this value represents the accuracy of the users in a best case scenario. Analysis of the accuracy of the

Table 3: Average duration for 24 selections.

	VirtualTablet	CapacitiveTablet	Controller	NonVRTablet
\emptyset	60,120	61,770	29,350	22,060
\pm	15,690	15,270	7,695	4,243

6 targets as seen in Figure 1 reveals, that the VirtualTablet detection has a higher accuracy of about 1 mm at the top row. The other techniques do not show such a large difference. Also, no learning effect can be detected in the collected data. Figure 13 shows the average distances to the center in a temporal order. The difference from the center stays approximately the same over the period of the selection task. The total amount of time needed to perform the selection task with each input technique is shown in Figure 11b and Table 3. Users took about twice as much time with the VirtualTablet and CapacitiveTablet in regards to the Controller interaction and almost three time as much in comparison to the NonVRTablet.

6.2 Tracing accuracy

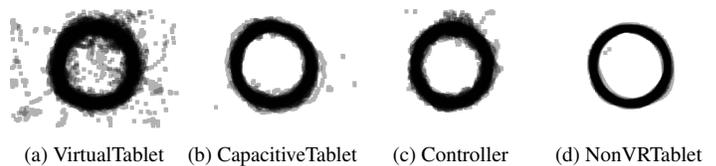


Figure 14: Stacked drawings of all circle traces.

Table 4: Shape tracing accuracy for the circle.

	VirtualTablet	CapacitiveTablet	Controller	NonVRTablet
% of drawing inside target area				
\emptyset	35,131	47,709	45,059	63,884
\pm	8,884	11,847	11,239	8,949
% of target area filled				
\emptyset	56,840	66,057	63,671	77,615
\pm	14,452	16,490	17,694	13,091
Average distance to target shape (mm)				
\emptyset	4,274	2,960	2,892	1,873
\pm	2,106	1,851	1,003	0,590
Drawing duration (s)				
\emptyset	15,688	8,128	6,455	5,277
\pm	8,540	2,877	3,542	2,378

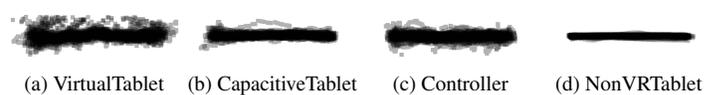


Figure 15: Stacked drawings of all line traces.

Table 5: Shape tracing accuracy for the line segment.

	VirtualTablet	CapacitiveTablet	Controller	NonVRTablet
% of drawing inside target area				
Ø	39,058	56,813	49,715	82,004
±	9,464	14,805	18,014	11,840
% of target area filled				
Ø	61,949	67,469	64,202	78,607
±	13,468	17,716	23,724	11,706
Average distance to target shape (mm)				
Ø	4,340	2,757	3,430	2,400
±	1,383	1,387	1,640	0,848
Drawing duration (s)				
Ø	6,885	3,909	3,148	2,521
±	4,075	1,424	2,179	1,172

The results for the shape tracing task are shown in Table 4 and 5 for the circle target and line target respectively. Drawing with the NonVRTablet achieves the highest tracing accuracy regarding percentage of drawing inside the target area and target area filled. The CapacitiveTablet is on average slightly better than the Controller and VirtualTablet condition. The standard deviation is smallest for the VirtualTablet and the NonVRTablet. The average distance towards the circle target area is very similar to this, with the NonVRTablet performing best with 1.9 mm accuracy and VirtualTablet performing worst with a distance of 4.3 mm. For the line segment, the average distance of the CapacitiveTablet is very close to the NonVRTablet with 2.8 mm to 2.4 mm. The detected touches of the VirtualTablet have the highest average distance (4.3 mm), but are not so far away from the Controller tracing (3.4 mm). Figures 14 and 15 show the shape traces of all users on top of each other. The touch detection of the VirtualTablet leads to a lot of smaller errors outside of the target area. The ground truth touch of the CapacitiveTablet is a lot more stable. The Controller drawings appear shaky. The NonVRTablet matches the target area the most.

Regarding shape tracing duration, the Controller is on average faster, but more inaccurate than the CapacitiveTablet. The NonVRTablet is the quickest and the VirtualTablet takes the most time.

6.3 Performance and User Experience

The results of the evaluation show that the arrangement of the extended menus with information displayed outside the interactable area was clear (see Table 6). It was also very useful to have more than one input surface. Teleportation with the tilting map has a medium difficulty. During the user study users often needed help to initially understand what they needed to do. Yet, once learned, participants quickly got better. Turning the valves with the orientation of the tablet and using the displayed buttons and sliders also has a medium difficulty. The pose of the tablet and touch input is recognized medium-well. The participants are happy with the interaction distance, which is due to the camera sensor at least 30 cm in front of the HMD. Input delay was quite high (mostly due to the

Table 6: 7 point Likert scale questionnaire for different aspects of the extended interaction techniques of the VirtualTablet.

Questions	Value 1	Value 7	Ø	±
Information outside the input area was...	unclear	clear	5,143	1,597
Using more than one input surface are was...	practical	impractical	2,429	1,116
Teleportation with the tilting map was...	easy	difficult	4,107	1,718
Using the tablet orientation as input was...	easy	difficult	3,607	1,800
Using the buttons and sliders was...	easy	difficult	3,643	1,315
Reaction of touch input was was...	expected	unexpected	3,786	1,319
Recognition of tablet pose was...	expected	unexpected	2,571	1,400
Minimum distance for input was...	too far away	just right	5,071	1,534
Input delay was...	annoying	undisturbing	3,786	1,839
Haptic surface for input was...	helpful	unnecessary	1,500	1,086

cameras), which is a little bit annoying for the users. However, the haptic surface for hand gesture input is clearly rated as helpful.

Figure 16 shows the results of the UEQ-S ratings. The pragmatic quality is very good for the Controller, positive for the CapacitiveTablet and neutral for the VirtualTablet (positive evaluation from a value of 0.8 and more). the hedonic quality is rated positive for all techniques, with the VirtualTablet and Controller rated best. The overall result shows that the Controller technique works very well. However, both touch surface interactions receive a positive rating. Compared to the supplied benchmark ratings from the UEQ Data Analysis Tool [23] the VirtualTablet is ranked as bad for the pragmatic quality, above average for the hedonic quality and below average overall.

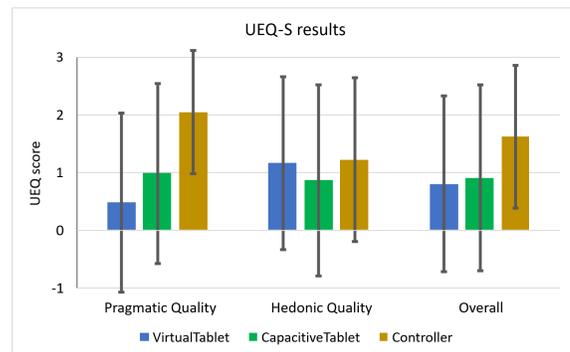


Figure 16: Results of UEQ-S.

The CapacitiveTablet is below average for all three categories. The Controllers are ranked as excellent for the pragmatic quality and overall. Their hedonic quality is above average. The difference between the Controller technique and the two touch techniques is significant for the pragmatic quality ($p \leq 0.001$) and overall

($p \leq 0.002$ and $p \leq 0.012$ for the difference towards the VirtualTablet and CapacitiveTablet respectively).

The results of the Raw NASA TLX as seen in Figure 17 show that the Controller ranks best in all categories. The VirtualTablet ranks worst in almost all categories, except for the physical demand.

Users rank the wearing comfort of the VirtualTablet and the Controllers as good, the CapacitiveTablet as below medium (see Figure 18). The quality of input is ranked best for the Controllers, good for the CapacitiveTablet and below medium for the VirtualTablet. Overall, the participants rank the Controller as good and the tablet techniques as medium.

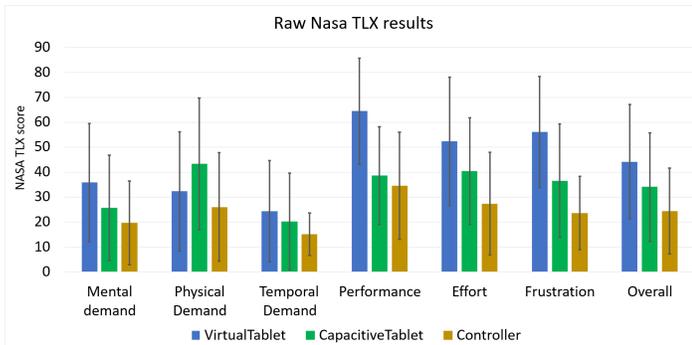


Figure 17: Results of Raw NASA TLX.

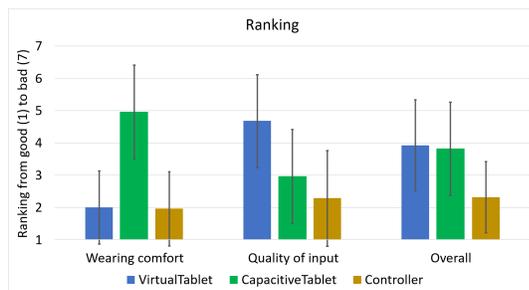


Figure 18: Rankings of the techniques.

7 Discussion

The rTOST test shows that both the VirtualTablet and the Controller interaction perform equally in a selection task. The precision of the VirtualTablet is about 6.9 mm, which is over half as small as the average size of a fingertip with 1620 mm in diameter [24]. The precision is good enough to interact with objects of suggested minimum target sizes, e.g. 9.6 mm [25] or 7-10 mm⁶. Because of the larger temporal delay, the tablet interaction is slower. This yields to longer interaction times and also affects the precision of the task execution, since the visualized hand and tablet locations did not match current location of their real world counterpart. The interaction with a controller is different compared to touch interaction. A user often only twists the wrist to point the laser towards another target. This is quicker than moving the whole hand, but it also induces more jitter as the larger standard deviations show. The shape tracing task

shows that the touch detection of VirtualTablet is not robust enough. Due to the large amount of noise and invalid data in the calculated depth information, the touch detection is not able to always detect a continuous touch and sometimes detects false positives. The touch detection also fails, if a user obscures the fingertip with her or his own hand. This explains the larger time difference and usability issues of the VirtualTablet compared to the CapacitiveTablet and Controller techniques, which recognize the touch through sensors.

VR does not allow the user to see her or his own hand. A good visualization of the user's hand is therefore necessary to achieve good results. The used point cloud and fingertip spheres depend on the segmentation of the hand. The blur during head movements and the automatic brightness adjustments dampen the quality of the image. However, the point cloud representation by itself seems to be not so easy to understand spatially. The impact of the hand visualization in the shape tracing task is lower than in the selection task for the CapacitiveTablet, because the user receives direct feedback from her or his touch point on the capacitive display and can compensate for any hand visualization delay or errors. Thus, the CapacitiveTablet performs better than the VirtualTablet.

The haptic surface for the interaction was rated as very helpful during the application task. This was also indicated through comments by the participants. The acrylic glass material is very lightweight (120 g) compared to the touchscreen tablet with battery (754 g) which leads to a lower physical demand and a higher wearing comfort (Vive controllers 2 × 203 g). The impact of the weight will be even larger at longer sessions of usage.

The comparable MRTouch system [12] shows slightly better selection precision ($\emptyset 5.4 \pm 3.2$ mm) when compared to the VirtualTablet. However, the VirtualTablet interaction uses movable touch surfaces, which induce additional precision errors. Furthermore, because MRTouch is an AR augmented reality (system, there is no need for a hand or surface visualization. But, the visualization in VR is crucial and has a larger impact on the precision and usability of the input technique. Tracing accuracy of VirtualTablet and MRTouch are comparable with an average distance of $\emptyset 4.0 \pm 3.4$ mm for MRTouch. Although the AR HMD of MRTouch uses a time of flight depth sensor and infrared cameras, the touch detection works with a threshold distance of 10 mm (ours 5 mm) which could lead to a touch recognition before the user reached the surface. The very accurately tracked Controllers show a similar target selection result as VirtualTablet and MRTouch. This shows that touch input in VR and AR is very accurate, even with the present issues regarding inaccurate tracking.

Our ground truth baseline (NonVRTablet with $\emptyset 2.3 \pm 1.4$ mm avg. distance to target) shows that there is room for improving the accuracy of the presented technique. Yet, the extended functionality, e.g. extending the user interface over the edges of the physical surface, turning the tablet around for a menu switch or changing the pose of the tablet to control a device, has a benefit that a real-world touch tablet cannot offer.

8 Conclusion and Future Work

We presented VirtualTablet, a movable surface that allows familiar touch interaction to control the VE. VirtualTablet is intuitive to

⁶see <https://support.google.com/accessibility/android/answer/7101858>

control, cheap, lightweight and can be shaped into any form and size. It can be used as an extension to currently supported input hardware, like controllers, since it does not require any additional hardware setup (except the board itself) and does not alter the VR HMD in any way. The haptic surface is helpful and the implementation allows an accurate selection of targets which is comparable to a laserpointer-based interaction using controllers.

The hardware used in this study has high latency and large noise levels. This yields to issues with continuous touch detection and dampens the overall accuracy of the system. For future work we would like to use a different camera and/or depth sensor to resolve these challenges. Also, other sensors like a capacitive touch foil on the tablet could be used to eliminate the problems of occlusion. We proposed several new interaction techniques that extend the usefulness of touch interaction in VR environments. With better tracking, further exploration of the design space of touch in VR will be possible. Another aspect that we need to improve is the hand visualization so that users get a better understanding of their finger positions. Mesh- or skeleton-like representations could be used for this purpose.

Conflict of Interest The authors declare no conflict of interest.

Acknowledgment We want to thank all of the participants of the user study for taking the time to provide us with very helpful feedback and insights.

References

- [1] A. H. Hoppe, F. Marek, F. van de Camp, and R. Stiefelwagen, "Virtualtablet: Extending movable surfaces with touch interaction", in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, Mar. 2019, pp. 980–981. DOI: 10.1109/VR.2019.8797993.
- [2] D. Markov-Vetter, E. Moll, and O. Staadt, "Evaluation of 3d selection tasks in parabolic flight conditions: Pointing task in augmented reality user interfaces", in *Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, ser. VRCAI 12, Singapore, Singapore: Association for Computing Machinery, 2012, 287294, ISBN: 9781450318259. DOI: 10.1145/2407516.2407583.
- [3] E. Koskinen, T. Kaaresoja, and P. Laitinen, "Feel-good touch: Finding the most pleasant tactile feedback for a mobile touch screen button", in *Proceedings of the 10th International Conference on Multimodal Interfaces*, ser. ICMI 08, Chania, Crete, Greece: Association for Computing Machinery, 2008, 297304, ISBN: 9781605581989. DOI: 10.1145/1452392.1452453.
- [4] C. S. Tzafestas, K. Birbas, Y. Koumpouros, and D. Christopoulos, "Pilot evaluation study of a virtual paracentesis simulator for skill training and assessment: The beneficial effect of haptic display", *Presence*, vol. 17, no. 2, pp. 212–229, Apr. 2008, ISSN: 1054-7460. DOI: 10.1162/pres.17.2.212.
- [5] S. Scheggi, L. Meli, C. Pacchierotti, and D. Prattichizzo, "Touch the virtual reality: Using the leap motion controller for hand tracking and wearable tactile devices for immersive haptic rendering", in *ACM SIGGRAPH 2015 Posters*, ser. SIGGRAPH 15, Los Angeles, California: Association for Computing Machinery, 2015, ISBN: 9781450336321. DOI: 10.1145/2787626.2792651.
- [6] A. L. Simeone, E. Velloso, and H. Gellersen, "Substitutional reality: Using the physical environment to design virtual reality experiences", in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI 15, Seoul, Republic of Korea: Association for Computing Machinery, 2015, 33073316, ISBN: 9781450331456. DOI: 10.1145/2702123.2702389.
- [7] B. Araujo, R. Jota, V. Perumal, J. X. Yao, K. Singh, and D. Wigdor, "Snake charmer: Physically enabling virtual objects", in *Proceedings of the TEI 16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*, ser. TEI 16, Eindhoven, Netherlands: Association for Computing Machinery, 2016, 218226, ISBN: 9781450335829. DOI: 10.1145/2839462.2839484.
- [8] L. Kohli, "Redirected touching: Warping space to remap passive haptics", in *2010 IEEE Symposium on 3D User Interfaces (3DUI)*, Mar. 2010, pp. 129–130. DOI: 10.1109/3DUI.2010.5444703.
- [9] 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*, ser. CHI 16, San Jose, California, USA: Association for Computing Machinery, 2016, 19681979, ISBN: 9781450333627. DOI: 10.1145/2858036.2858226.
- [10] L.-P. Cheng, E. Ofek, C. Holz, H. Benko, 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*, ser. CHI 17, Denver, Colorado, USA: Association for Computing Machinery, 2017, 37183728, ISBN: 9781450346559. DOI: 10.1145/3025453.3025753.
- [11] D. Medeiros, L. Teixeira, F. Carvalho, I. Santos, and A. Raposo, "A tablet-based 3d interaction tool for virtual engineering environments", in *Proceedings of the 12th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, ser. VRCAI 13, Hong Kong, Hong Kong: Association for Computing Machinery, 2013, 211218, ISBN: 9781450325905. DOI: 10.1145/2534329.2534349.
- [12] R. Xiao, J. Schwarz, N. Throm, A. D. Wilson, and H. Benko, "Mrtouch: Adding touch input to head-mounted mixed reality", *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 4, pp. 1653–1660, Apr. 2018, ISSN: 2160-9306. DOI: 10.1109/TVCG.2018.2794222.
- [13] S. Garrido-Jurado, R. Muoz-Salinas, F. Madrid-Cuevas, and M. Marn-Jimenez, "Automatic generation and detection of highly reliable fiducial markers under occlusion", *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014, ISSN: 0031-3203. DOI: 10.1016/j.patcog.2014.01.005.
- [14] N. S. Chethana, Divyaprabha, and M. Z. Kurian, "Design and implementation of static hand gesture recognition system for device control", in *Emerging Research in Computing, Information, Communication and Applications*, N. R. Shetty, N. H. Prasad, and N. Nalini, Eds., Singapore: Springer Singapore, 2016, pp. 589–596, ISBN: 978-981-10-0287-8.
- [15] D. C. Niehorster, L. Li, and M. Lappe, "The accuracy and precision of position and orientation tracking in the htc vive virtual reality system for scientific research", *i-Perception*, vol. 8, no. 3, 2017. DOI: 10.1177/2041669517708205.
- [16] I. Poupyrev, T. Ichikawa, S. Weghorst, and M. Billingham, "Egocentric object manipulation in virtual environments: Empirical evaluation of interaction techniques", *Computer Graphics Forum*, vol. 17, no. 3, pp. 41–52, 1998. DOI: 10.1111/1467-8659.00252.
- [17] C. Harrison, H. Benko, and A. D. Wilson, "OmniTouch: Wearable multitouch interaction everywhere", in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST 11, Santa Barbara, California, USA: Association for Computing Machinery, 2011, 441450, ISBN: 9781450307161. DOI: 10.1145/2047196.2047255.
- [18] S. G. Hart, "Nasa-task load index (nasa-tlx); 20 years later", *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 50, no. 9, pp. 904–908, 2006. DOI: 10.1177/154193120605000909.
- [19] M. Schrepp, A. Hinderks, and J. Thomaschewski, "Design and evaluation of a short version of the user experience questionnaire (ueq-s)", *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 4, p. 103, Jan. 2017. DOI: 10.9781/ijimai.2017.09.001.
- [20] K. K. YUEN, "The two-sample trimmed t for unequal population variances", *Biometrika*, vol. 61, no. 1, pp. 165–170, Apr. 1974, ISSN: 0006-3444. DOI: 10.1093/biomet/61.1.165.
- [21] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other", *The Annals of Mathematical Statistics*, vol. 18, no. 1, pp. 50–60, 1947, ISSN: 00034851. DOI: 10.1214/aoms/1177730491.

- [22] J. Cohen, *Statistical power analysis for the behavioral sciences. (2nd ed.)* 1988.
- [23] A. Hinderks, M. Schrepp, and J. Thomaschewski, "A benchmark for the short version of the user experience questionnaire", Sep. 2018. DOI: 10.5220/0007188303730377.
- [24] K. Dandekar, B. I. Raju, and M. A. Srinivasan, "3-D Finite-Element Models of Human and Monkey Fingertips to Investigate the Mechanics of Tactile Sense", *Journal of Biomechanical Engineering*, vol. 125, no. 5, pp. 682–691, Oct. 2003, ISSN: 0148-0731. DOI: 10.1115/1.1613673.
- [25] P. Parhi, A. K. Karlson, and B. B. Bederson, "Target size study for one-handed thumb use on small touchscreen devices", in *Proceedings of the 8th Conference on Human-Computer Interaction with Mobile Devices and Services*, ser. MobileHCI 06, Helsinki, Finland: Association for Computing Machinery, 2006, 203210, ISBN: 1595933905. DOI: 10.1145/1152215.1152260.