Enhancing Touch Input in Desktop VR: A Hybrid Approach Combining Headset-Based Hand Tracking and Touch Detection with a Smartphone

Taiga Kashima
Preferred Networks Inc.
Tokyo, Japan
tkashima@preferred.jp

Daichi Suzuo
Preferred Networks Inc.
Tokyo, Japan
suzuo@preferred.jp

Fabrice Matulic
Preferred Networks Inc.
Tokyo, Japan
fmatulic@preferred.jp

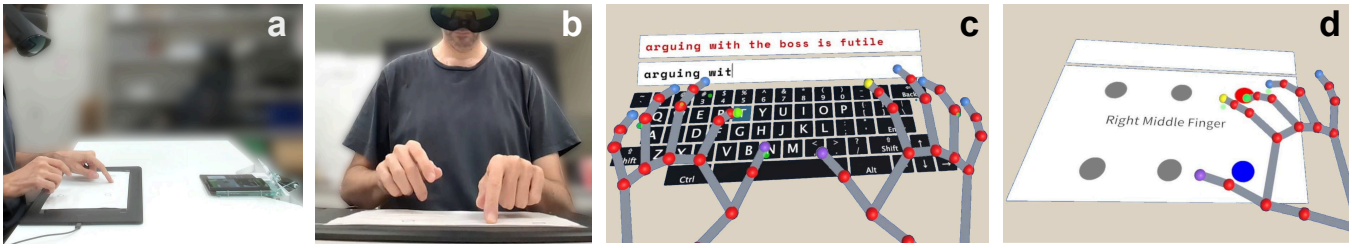

Figure 1: Touch detection on ordinary surfaces using a mobile phone for desktop VR. Phone placed vertically upside down so that its camera captures the user's hand and finger touches at a low angle (a); View from the phone camera (b); VR hands tracked by the headset with touch input detected by the phone for typing (c) and tapping (d).

Abstract

Leveraging everyday surfaces for touch input in virtual reality (VR) presents challenges due to the insufficient precision of headset-based hand tracking for reliable surface contact detection. To enhance touch input, we introduce a novel approach integrating headset-based hand tracking with touch detected by a smartphone placed on the surface. The smartphone's camera captures hand movements from a low angle, and a neural network identifies finger contacts, which are correlated with the fingers tracked by the headset. Evaluations using a Quest Pro and a Wacom tablet in text-typing and tapping tasks show that both phone-based and tablet-based touch detection improve performance to varying degrees, but neither technique fully overcomes the limitations of hand tracking.

CCS Concepts

• **Human-centered computing** → **Virtual reality**; **Text input**;
• **Computing methodologies** → *Computer vision.*

Keywords

Virtual Reality, Hand Tracking, Touch Detection, Text Entry

1 Introduction

Virtual reality (VR) is increasingly used for everyday computing, leading to a growing interest in enhancing interaction modalities for such tasks. One promising avenue leverages ubiquitous surfaces like desktops and tables as interactive planes for touch input within immersive 3D virtual environments. While recent VR headsets with hand tracking capabilities offer the potential for surface interaction, their precision remains insufficient for reliable touch detection and falls short of the experience offered by conventional touchscreens.

To address these shortcomings, research efforts have explored a variety of external sensor setups to enhance touch detection accuracy [3, 8, 10, 15, 17, 19, 21, 29, 32, 33]. However, these systems often introduce practical challenges, such as complex setups and potential user inconvenience when sensors have to be worn. Furthermore, evaluations of these systems have primarily focused on touch classification accuracy without assessing the possible impact on the holistic touch experience in VR or comparing performance against readily available techniques like headset-based hand tracking and common touch devices.

We present a novel approach that integrates headset-based hand tracking with touch detection using images captured by the camera of a smartphone placed on the surface (Figure 1). We conduct tapping and typing experiments using a Quest Pro to evaluate this phone-based technique against a Wacom tablet and pure headset hand tracking. Our results demonstrate that headset-only hand tracking remains inadequate for reliable touch input. While both our phone-based approach and the tablet improve performance, neither fully overcomes the inherent limitations of headset-based hand tracking. We analyse the trade-offs and challenges associated with fusing headset-based hand tracking and external touch sensing, and offer insights for the design and development of future touch interfaces in desktop VR environments.

2 Related Work

Transforming ordinary surfaces into interactive touch interfaces has been a long-standing research goal. In non-XR contexts, this

CHI EA '25, Yokohama, Japan

ACM ISBN 979-8-4007-1395-8/25/04
https://doi.org/10.1145/3706599.3719768

has been achieved through various methods combining projectors with diverse touch-sensing technologies, including depth cameras [32, 33], multiple RGB cameras [2, 13], infrared [3], piezoelectric [22], acoustic [12] and vibration [20, 26] sensors. Further approaches exploit optical distortions in structured light and projected content [4, 11, 31] or use wearable sensors [10, 18, 27, 35] to detect touch input.

Within XR, modern headsets with integrated hand tracking and surface detection enable basic touch interaction without external sensors [6, 34]. However, these methods suffer from precision and latency limitations that make them less practical for high-precision and high-frequency touch tasks. For instance, Meta Quest devices exhibit an average positional error exceeding 1cm for tracked fingers [1, 25]). To enhance touch detection in XR, researchers have used external sensors similar to those used in non-XR settings, such as infrared thermographic camera [14], remote vibrometry [29], shadows cast by light sources mounted on the user's wrists [15] and wearable IMUs [8, 17, 19].

While these external sensor systems can improve accuracy and response times, they often require instrumenting the user, the environment, or the headset with specialised hardware. Furthermore, evaluations of those techniques have mainly focused on touch classification accuracy using custom test datasets and targeting precision has not been compared head-to-head to baselines that use only headset-based hand tracking, which is becoming increasingly common in XR. Our approach does not instrument the user or the headset and uses only an unmodified smartphone to enhance touch interaction. Furthermore, we compare those techniques with pure headset-based hand tracking as well as a touch-sensitive tablet, which allows us to demonstrate the potential benefits of supplementing standard VR setups with ubiquitous devices for touch sensing.

3 Phone-Based Touch Detection

3.1 Approach

Our approach combines hand tracking from the headset with external touch detection via a smartphone placed on the desktop surface in front of the touch input area so that the phone's camera captures the user's hands from a low angle, almost parallel to the surface. This setup can be achieved by placing the phone vertically upside down (e.g., stabilised using its protective sleeve), or, as shown in Figure 1a, placing it flat with a reflector such as a prism [37], a mirror [16] or an omni-directional lens [36] attached over the front camera to redirect the optical path. Because the camera view is not perfectly aligned with the surface plane, a slight perspective distortion remains, making it impractical to use more straightforward computer vision approaches like defining a horizontal line on the captured images and triggering touch events when fingertips cross that line. Moreover, our ultimate goal is to allow users to simply place the phone on the desk and immediately use it for touch detection without any prior calibration. To achieve accurate detection under these constraints, we opt for a deep-learning-based approach.

3.2 Deep Neural Network Combined with Tracked Hands

We create and train a neural network based on a MobileNet backbone [9], which outputs a set of probabilities of each finger from both hands touching the surface (full detail of the network in the Annex). This information is integrated with hand tracking provided by the VR headset. Specifically, touch events are triggered based on the network's predictions when a probability exceeds a specific threshold, regardless of whether the fingertip tracked by the headset actually collides with the virtual touch plane aligned with the physical surface. To determine the touch point, the tip of the corresponding finger of the virtual hand is projected onto the plane and this projection point is used as the touch point. Green markers representing those projected points are shown when fingertips approach the surface to help users aim more precisely. Hand or fingers are not repositioned upon contact like Zhu et al's dynamic recalibration [39], as we find this causes excessive disruption for the user, especially in high-frequency touch tasks.

3.3 Implementation

3.3.1 Model Performance. We assessed the performance of our model on a dataset consisting of 5400 images from 16 participants performing mock typing and tapping gestures using leave-one-out cross-validation. We obtained detection accuracies ranging from 99.3% to 99.9% (see full details in the Appendix), suggesting very high detection accuracy.

3.3.2 Model Deployment. As its name suggests, MobileNet is optimised for deployment on mobile devices, and we created a model to deploy on our Pixel 7 using TorchScript. We developed an Android application that captures images from either the front or rear camera, crops them to the required size and performs inference to detect touches. However, our Android MobileNet achieved a performance of 4 fps, which is insufficient for high-frequency touch detection. We therefore deployed our model on a server with a Geforce RTX 3090 and streamed images from the phone over WiFi. This setup enabled us to achieve a stable inference speed of 30 fps.

3.3.3 VR Client. The VR client, which receives the estimated touch status of each finger and integrates this data with hand tracking, was developed in Unity. The client ran on a Meta Quest Pro, which uses Meta Hand Tracking 2.1.

As a second baseline for comparison with our approach, we integrated a Wacom Intuos tablet into our setup, leveraging its reliable capacitive touch sensing as a reference point. The tablet was placed on the desk and connected to the server via USB. Due to the tablet's inability to directly identify individual fingers, we determined the touching finger by proximity to the virtual fingertip. Spatial correspondence between the tablet's touch surface and the VR environment was established by manually positioning the VR view to match three virtual markers with physical landmarks on the tablet. This manual calibration was necessary due to the Quest's incapability of tracking fiducial markers and lack of developer access to its camera, which prevent automatic calibration.

We used the tablet to estimate the latency of touch detection through the phone and the server, specifically by calculating the difference between the timestamps of touch points registered by the

phone-based system and those detected by the tablet. We obtained an average latency of 88ms.

4 Evaluation

We conducted experiments to evaluate the effectiveness of our hybrid touch detection technique.

4.1 Design

Our evaluation adopted a within-subjects design and consisted of two types of tasks performed in VR while sitting at a desk: text-typing, which is a common task on touch interfaces that involves frequent touch input, and tapping, which focuses on targeting speed and precision with specific fingers.

We compared our phone-based technique with touch input detected by a tablet and pure hand tracking from the Quest headset, i.e. three conditions: Phone, Tablet, Quest. All conditions used Quest hand tracking to render the hands in VR using a stick hand model (Figure 1c and d). The Quest conditions handled touch input detection by identifying fingertips of the VR hand model that collided with the virtual plane aligned with the desk surface.

4.1.1 Text Entry Experiment. Previously proposed text entry techniques on flat surfaces combining hand tracking or touch sensing with language or typing models achieve high performance [7, 23, 24, 28, 30], but the reliance on language models limits the applicability of these techniques to input of natural phrases in the language supported by the models and their underlying dictionaries. Our goal was to assess text entry performance using pure touch detection, which is more generalisable.

For the tasks we used the TextTest++ tool [38], which shows sequences of English phrases for the participant to transcribe. From the total phrase set of TextTest++, we constructed three subsets of five phrases with each subset containing 30 words and 150 characters to ensure balance. The task consisted in correctly typing the prompt phrases one after the other with a virtual keyboard (Figure 1c), correcting any mistake to ensure transcribed phrases matched the prompts. The width of each letter key of the virtual keyboard was 1.8 cm and that of the backspace key was 3.9 cm. Text entry performance was determined by the number of words typed per minute (which included the time for corrections) and the corrected error rate (rate of incorrectly typed characters relative to the total number of characters entered).

4.1.2 Tapping Experiment. The tapping experiment was a targeting task, in which participants were required to successively tap circle targets arranged on a 3×2 grid (Figure 1d). There were two target sizes: large targets with a diameter of 2.4 cm, which approximately corresponds to the width of a key on the Meta Quest virtual keyboard, and small targets with a diameter of 1.4 cm, roughly the width of an icon on the Quest toolbar. The task sequence began with tapping a grid of large targets, followed by a grid of small targets.

Each target in the grid had to be tapped twice in vertical back-and-forth motions within each column (distance between target pairs = 9.7 cm), except for the first target of each column (the starting point), which was tapped an additional time. Thus, for a set of targets of a specific size, participants performed 15 trials (5 targets × 3 columns), where targeting times were measured for the taps after the initial one in each column, which resulted in 12 measurements per set of targets (4 measurements × 3 columns).

Each target in a set had to be tapped with a specific finger of each hand in the following order: first the thumb, then the index finger, and finally the middle finger. The ring finger and pinkie were excluded from this study due to particularly low tracking precision revealed during pilot tests and their limited use for tapping. Participants performed the task using the fingers of their right hand first and then switched to their left hand.

During the task, the current target appeared in red while the subsequent target was shown in blue. In addition to the green fingertip projections on the surface to help them aim, participants were provided auditory feedback to guide their performance: a buzzer sound was emitted when a target was missed, and a beep sound was played if a non-designated finger was detected as having contacted the surface (possibly mistakenly). Participants proceeded to the next target if it was hit, or if it was missed within a 10 cm distance (classified as a near miss). Taps exceeding the 10 cm miss distance, however, were considered likely inadvertent or misdetections and the target was not validated in that case.

The total number of targets participants had to validate summed up to 180 (15 targets × 2 sizes × 3 fingers × 2 hands), resulting in 144 recorded targeting times (12 times × 2 sizes × 3 fingers × 2 hands). In addition to timestamps, we recorded wrong finger touches. We further logged all touch events for the currently tested technique, with the exception of tablet touches, which were logged in all conditions to serve as ground truth for tap detection accuracy analysis.

4.2 Participants

We recruited 18 participants (11 males, 7 females) of average age 35.7 years (SD=9.5) from our institution. Three had no prior VR experience, thirteen had had occasional previous exposure to VR in previous studies and exhibits, while two were regular users. Two participants had provided image data to train the model used for the Phone technique.

The order of conditions was counterbalanced among participants to minimise learning effects. Consequently, each phrase set in the typing task was used an equal number of times across all conditions. To prevent potential biases, participants were not informed of the specific techniques they were experiencing during the study. For the typing task, participants were not given any advice on typing styles that might be more efficient for the different techniques. This was to avoid influencing their approach and allow them to naturally discover effective strategies for each technique.

4.3 Procedure

The experiment setup consisted of a desk on which a Wacom Intuos Pro tablet and a Pixel 7 smartphone were placed, as shown in Figure 1a. We used a mirror reflecting the Pixel 7's front camera so that the phone could remain in a stable position while operating it during the experiments. The phone camera was at a 35cm distance from the Wacom tablet, which ensured that hands interacting with the virtual keyboard and the tapping panel were fully visible in the captured frames.

The procedure for each participant was as follows: After explaining the purpose and protocol of the study, the participant sat at the desk, put on the VR headset and adjusted it so that the VR scene appeared clearly. Using the arrow keys of a physical keyboard placed on their lap, the participant was given the opportunity to apply a height offset to the VR hands in order to closely align the real and virtual fingertip contact points on their respective surfaces, although perfect alignment was not possible due to mismatches between real and virtual fingers. This adjustment step was most important for the Quest condition, as it relied on the collision of the virtual fingertips with the plane to detect touches.

Participants started the typing task with the first technique, followed by the tapping task. Time to train was given before each task to allow participants to familiarise themselves with the touch response and precision of the technique. Upon completing both tasks with one technique, participants moved on to the next technique and repeated the process. After completing the tasks with all three techniques, participants were asked to rate the techniques on the last three scales of the NASA-TLX, i.e. performance, effort and frustration. We selected this subset of scales in consideration of participants' fatigue after performing six tasks and because effort covers physical and mental demand to a large extent. Then, participants were invited to provide reasons for their ratings as well as general feedback about their experience.

A full study session with one participant lasted approximately one hour and participants were rewarded with snacks.

5 Results

We analysed the study data by conducting repeated measures ANOVAs and post hoc tests when main effects were obtained to identify statistically significant differences. When the distribution was non-normal, we applied Aligned Rank Transform. Greenhouse-Geisser corrections were used if sphericity was violated, and Bonferroni corrections for the pairwise comparisons. Statistically significant differences are denoted by asterisks in figures (**** for $p \leq 0.0001$, *** for $p \leq 0.001$, ** for $p \leq 0.01$ and * for $p \leq 0.05$). Error bars represent 95% confidence intervals. A summary of the main results is provided below, with comprehensive statistical analyses detailed in the Annex.

5.1 Typing Task

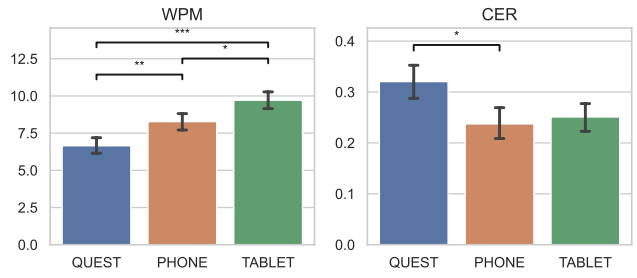

Figure 2: Words per minute (WPM) and corrected error rates (CER) for the typing task

Figure 2 shows the words per minute and error rates achieved by participants in the typing task. On average, participants reached speeds of 6.6 WPM with Quest ($SD = 2.7$), 8.3 WPM with Phone ($SD = 2.8$) and 9.7 WPM ($SD = 2.8$) with Tablet, with all differences statistically significant ($p = 0.002$ for Phone vs Quest, $p = 0.018$ for Phone vs Tablet and $p < 0.001$ for Quest vs Tablet). Note that these speeds include the time taken to correct errors, which more realistically reflects real-world typing performance than evaluations solely measuring raw input speed. Regarding error rates, participants made the most errors when using the Quest, with an average error rate of 0.32 ($SD = 0.16$). The rates for Phone (0.24, $SD = 0.15$) and Tablet (0.25, $SD = 0.14$) were comparable, however only the difference between Quest and Phone was statistically significant ($p = 0.019$).

These results suggest that typing performance is generally low across all conditions in this VR context. By comparison, Dudley et al. reported typing speeds exceeding 50 WPM on surface-aligned VR keyboards[5]. However, their study used a professional motion capture system for hand and finger tracking. This difference shows the substantial impact of headset hand-tracking on typing performance, even when a highly accurate touch sensor like a capacitive tablet is used.

5.2 Tapping Task

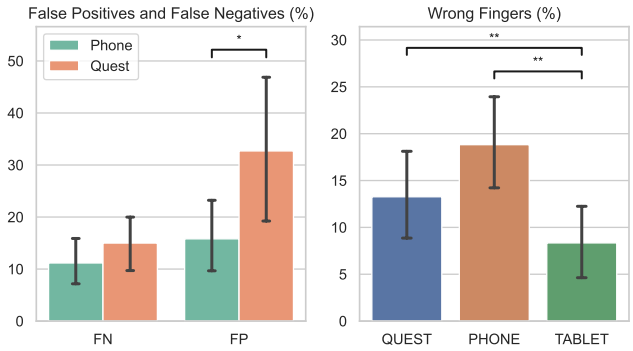

Figure 3: Percentage of false positives (FP), false negatives (FN) and detected wrong fingers in the tapping task

The tapping task focused on the accuracy of touch input from specific fingers, allowing us to examine touch detection accuracy and finger identification in greater detail.

5.2.1 Touch Detection Accuracy. We first analysed the touch detection accuracy of Quest and Phone (Figure 3), using touches recorded by the tablet as the ground truth. We considered the percentage of false negatives (FN) and false positives (FP) for both techniques. On average, Phone had 11.2% FN ($SD = 9.0$) and Quest had 15.0% FN ($SD = 10.4$). Phone had 15.8% FP ($SD = 13.5$) compared to 32.7% FP ($SD = 29.7$) for Quest, with only the difference in FP being statistically significant ($p = 0.008$). The high rate of FP for Quest is likely due to participants lowering the virtual hands during calibration to ensure all fingers triggered touch events upon physical contact with the desk. This adjustment led to multiple incorrect touch events being triggered when fingers were only hovering above the surface.

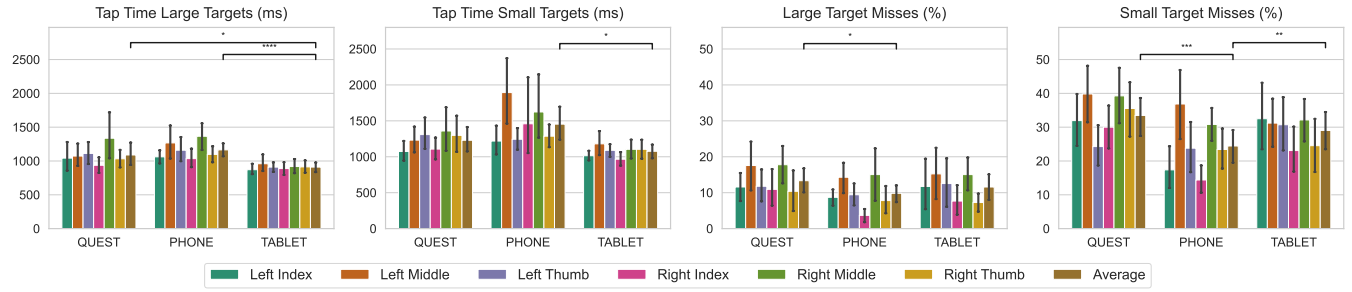

Figure 4: Tapping time between targets and number of target misses for each finger and target size.

5.2.2 Wrong Finger Touches. Regarding incorrect finger identification and touches, Phone had the highest error rate on average (18.8%, $SD = 10.9$), followed by Quest (13.2%, $SD = 10.0$), although this difference was only slightly above the threshold of statistical significance ($p = 0.0613$). Tablet had the lowest percentage of incorrect finger touches (8.3%, $SD = 8.8$), which was significantly lower than the other two conditions ($p = 0.002$ for both comparisons). This nonzero error rate for Tablet shows again the impact of hand tracking inaccuracies, which sometimes caused the incorrect finger to be identified.

5.2.3 Tapping Performance. Figure 4 shows tapping times between two consecutive targets in the same column and the number of target misses for the two target sizes. ANOVAs on Technique × Finger did not reveal any interaction effects in all cases, so we analysed the factors independently.

Participants generally tapped faster and at a more consistent pace with Tablet compared to the other conditions. For large targets, average tapping speed was 910ms for Tablet ($SD = 196$), which is significantly faster than Quest (1089ms, $SD = 442$) ($p = 0.038$) and Phone (1164ms, $SD = 352$) ($p < 0.0001$). For small targets, only Tablet (1076ms, $SD = 253$) was significantly faster than Phone (1454ms, $SD = 809$) ($p = 0.0111$), while Quest averaged 1229ms ($SD = 456$).

Regarding precision, participants were more accurate with Phone (24% misses, $SD = 15.6$) on small targets than Tablet (29%, $SD = 15.9$) and Quest (33%, $SD = 16.4$). The two-way ANOVA on Technique and Finger revealed interaction effects between the two variables and we refer to the Annex for the detailed statistical analyses of simple main effects. For large targets, participants made significantly fewer misses with Phone (9.8%, $SD = 9.1$) than Quest (13.3%, $SD = 11.1$) ($p = 0.0236$). The miss rate for large targets with Tablet was 11.6% ($SD = 11.5$) and differences with miss rates of other techniques were not statistically significant ($p = 0.248$ for both comparisons).

Accuracy varied across fingers, with participants showing greater precision using their index fingers than middle fingers. With Phone, they only missed small targets 5% and large targets 4% of the time with the index finger of their dominant hand, while achieving respectable speeds of 1s for large targets. Middle fingers were the least accurate with close to 40% misses on average for small targets.

5.2.4 Participant Ratings. Participants' ratings generally mirrored performance metrics. All techniques were rated above the midpoint value of 5, indicating somewhat challenging experiences.

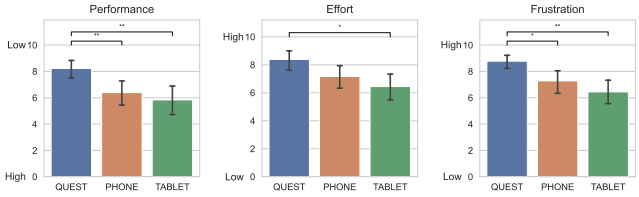

Figure 5: Participant ratings of their performance, effort and frustration.

Quest was the least favoured technique, with significantly lower performance ratings than Phone and Tablet ($p = 0.0016$ for both comparisons) and significantly higher frustration levels compared to both Phone ($p = 0.0007$) and Tablet ($p < 0.0001$). There were no significant differences between Phone and Tablet for these two metrics. For effort, Quest required significantly higher effort than Tablet ($p = 0.013$), with no other significant differences.

5.2.5 Participant Feedback. Qualitative feedback confirmed that many participants experienced difficulties with Quest, citing issues such as the need to press deeply for touch detection, particularly with certain fingers like the left thumb and middle finger, which affected accuracy. The technique required participants to calibrate hand height carefully, and errors were common if the calibration was not precise. Phone was often perceived as having marginally higher accuracy than Quest, especially for typing tasks. However, challenges persisted, particularly with the recognition of keys around the edges of the interaction area, and identification of middle fingers. Despite this, some participants expressed that this technique enabled them to type with fewer errors, albeit sometimes using compensatory strategies, such as using one or two fingers only. Tablet was reported to be more responsive overall. Rapid key presses could be reliably detected, although unintentional inputs also occurred.

Across all techniques, participants highlighted issues with hand tracking reliability, particularly for the left hand and middle fingers, which did not always match their real finger movements. Many participants also reported physical fatigue, particularly due to the headset and the need to adjust hand postures for fingers to be correctly tracked and touches properly recognised.

6 Discussion

The initial evaluation of our touch detection model on the training dataset demonstrated very high accuracy. However, in practice,

when integrated with headset hand tracking, touch input performance could only be modestly enhanced with our Phone technique or a Tablet. These results show the significant influence of the limitations of headset-based hand tracking technology. Previous work often sidestepped these issues by focusing narrowly on touch event classification or using advanced solutions like motion capture systems to track hands. *Our research exposes the challenges of using headset-based hand tracking to support touch input, even when augmented with external touch sensors.*

6.1 Participant Adaptation and Experience

None of the participants had prior experience typing on a hard surface in VR. This unfamiliarity required them to devise strategies to type as efficiently as possible given the imperfections of hand tracking and touch detection. Many initially tried to mimic their typing style on a physical or touch keyboard but quickly realised that this approach was ineffective in this context. Participants frequently cited the difficulty of overcoming ingrained expectations from their experience with physical and touchscreen keyboards. This disconnect between anticipated and actual performance likely contributed to the relatively low subjective experience ratings.

However, this may be a matter of familiarity and practice. One of the co-authors, who had ample time to acclimate to the techniques, can comfortably achieve over 25 WPM with Phone and over 30 WPM with Tablet (compared to ~21 WPM with Quest). Our study's one-hour duration provided insufficient time for participants to fully adapt to the VR keyboard and the headset's approximate hand tracking. As a result, a third of our participants opted to type with their index fingers only.

Generally, index fingers were the most accurately tracked by the headset and had the highest detection accuracy with Phone. This aligns with existing VR interfaces, which prioritise index fingers for pointing and mid-air touches, with many VR keyboards limiting input to index fingers only. Our findings suggest that index fingers should also be prioritised for touch input in desktop VR settings if hands cannot be tracked with very high precision. In such single-touch, index finger-based scenarios, camera or tablet-based touch input detection can offer noticeable benefits.

6.2 Limitations and Future Work

While our goal was to deploy the touch detection model directly on the phone, inference on our Pixel 7 phone was limited to 4fps. However, newer devices with dedicated AI chips are expected to enable significantly faster inference speeds. Additionally, we have only tested the network extensively on this phone model in a relatively controlled environment. Future work should capture diverse datasets from various environments, multiple devices, and a wide range of people to improve generalisability.

Hand tracking inconsistencies for different fingers, positions, and viewing angles were a notable limitation. The headset's cameras do not always have a clear view of the fingertips, resulting in unstable and inconsistent positions of the virtual hands and unreliable touch detection. This problem is inherently difficult to solve given the limited viewing angles of headset-embedded cameras. Potential solutions could involve additional environmental cameras for hand tracking to cover the blind spots of the headset's cameras alongside touch sensors [13], but this may introduce additional sensing and possibly calibration requirements, unlike our streamlined single-phone approach.

While a phone's camera with a frontal view of the hands is ideal for tracking single touches, occlusions can occur for multiple touches on the same optical axis of the camera, such as when performing two-finger rotations. Additionally, hands moving out of the camera's field of view would no longer be tracked. The latter issue could be mitigated by attaching a wide-angle or fisheye lens to the phone's camera to increase the capture view and thus maintain continuous tracking.

One way to bypass these limitations entirely when using a tablet or other 2D touch sensing devices is to allow users to see their real hands directly via passthrough, thus eliminating the need for hand tracking. However, this solution can introduce disruptions to the fully immersive VR experience with potential visual artefacts from the camera interfering with the virtual touch UI. This approach also precludes the use of customised VR hand representations.

7 Conclusion

We introduced a hybrid approach to enhance touch input in desktop VR by combining headset-based hand tracking with touch detection via a smartphone placed on the desk. Our method utilises the smartphone's camera and a deep neural network to identify finger contacts, which are synchronised with headset-tracked fingers. Evaluations in text-typing and tapping tasks comparing our phone-based touch detection technique with a tablet and pure headset-tracked fingers show that the two external sensing techniques can improve touch input performance, but they are constrained by the limitations of headset hand tracking. We hope our findings will inform and inspire future research to create more robust and intuitive touch interfaces for desktop VR.

References

[1] Diar Abdlkarim, Massimiliano Di Luca, Poppy Aves, Mohamed Maaroufi, Sang-Hoon Yeo, R Chris Miall, Peter Holland, and Joeseph M Galea. 2024. A methodological framework to assess the accuracy of virtual reality hand-tracking systems: A case study with the Meta Quest 2. *Behavior research methods* 56, 2 (2024), 1052–1063.

[2] Ankur Agarwal, Shahram Izadi, Manmohan Chandraker, and Andrew Blake. 2007. High precision multi-touch sensing on surfaces using overhead cameras. In *Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP'07)*. IEEE, 197–200.

[3] Alex Butler, Shahram Izadi, and Steve Hodges. 2008. SideSight: multi-"touch" interaction around small devices. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology* (Monterey, CA, USA) *(UIST '08)*. Association for Computing Machinery, New York, NY, USA, 201–204. doi:10.1145/1449715.1449746

[4] Jingwen Dai and Chi-Kit Ronald Chung. 2014. Touchscreen Everywhere: On Transferring a Normal Planar Surface to a Touch-Sensitive Display. *IEEE Transactions on Cybernetics* 44, 8 (2014), 1383–1396. doi:10.1109/TCYB.2013.2284512

[5] John Dudley, Hrvoje Benko, Daniel Wigdor, and Per Ola Kristensson. 2019. Performance Envelopes of Virtual Keyboard Text Input Strategies in Virtual Reality. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 289–300. doi:10.1109/ISMAR.2019.00027

[6] Camille Dupré, Caroline Appert, Stéphanie Rey, Houssem Saidi, and Emmanuel Pietriga. 2024. TriPad: Touch Input in AR on Ordinary Surfaces with Hand Tracking Only. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 754, 18 pages. doi:10.1145/3613904.3642323

[7] Xingyu Fu and Mingze Xi. 2024. Typing on Any Surface: Real-Time Keystroke Detection in Augmented Reality. In *2024 IEEE International Conference on Artificial Intelligence and eXtended and Virtual Reality (AIxVR)*. 350–354. doi:10.1109/AIxVR59861.2024.00060

[8] Yizheng Gu, Chun Yu, Zhipeng Li, Weiqi Li, Shuchang Xu, Xiaoying Wei, and Yuanchun Shi. 2019. Accurate and Low-Latency Sensing of Touch Contact on Any Surface with Finger-Worn IMU Sensor. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 1059–1070. doi:10.1145/3332165.3347947

[9] A. Howard, M. Sandler, B. Chen, W. Wang, L. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le. 2019. Searching for MobileNetV3. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, Los Alamitos, CA, USA, 1314–1324. doi:10.1109/ICCV.2019.00140

[10] Min-Chieh Hsiu, Chiuan Wang, Da-Yuan Huang, Jhe-Wei Lin, Yu-Chih Lin, De-Nian Yang, Yi-ping Hung, and Mike Chen. 2016. Nail+: sensing fingernail deformation to detect finger force touch interactions on rigid surfaces *(MobileHCI '16)*. Association for Computing Machinery, New York, NY, USA, 1–6. doi:10.1145/2935334.2935362

[11] Jun Hu, Guolin Li, Xiang Xie, Zhong Lv, and Zhihua Wang. 2014. Bare-fingers Touch Detection by the Button's Distortion in a Projector–Camera System. *IEEE Transactions on Circuits and Systems for Video Technology* 24, 4 (2014), 566–575. doi:10.1109/TCSVT.2013.2280088

[12] Yasha Iravantchi, Yi Zhao, Kenrick Kin, and Alanson P. Sample. 2023. SAWSense: Using Surface Acoustic Waves for Surface-bound Event Recognition. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 422, 18 pages. doi:10.1145/3544548.3580991

[13] Itai Katz, Kevin Gabayan, and Hamid Aghajan. 2007. A Multi-touch Surface Using Multiple Cameras. In *Advanced Concepts for Intelligent Vision Systems*, Jacques Blanc-Talon, Wilfried Philips, Dan Popescu, and Paul Scheunders (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 97–108.

[14] Daniel Kurz. 2014. Thermal touch: Thermography-enabled everywhere touch interfaces for mobile augmented reality applications. In *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 9–16. doi:10.1109/ISMAR.2014.6948403

[15] Chen Liang, Xutong Wang, Zisu Li, Chi Hsia, Mingming Fan, Chun Yu, and Yuanchun Shi. 2023. ShadowTouch: Enabling Free-Form Touch-Based Hand-to-Surface Interaction with Wrist-Mounted Illuminant by Shadow Projection. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) *(UIST '23)*. Association for Computing Machinery, New York, NY, USA, Article 27, 14 pages. doi:10.1145/3586183.3606785

[16] Fabrice Matulic, Aditya Ganeshan, Hiroshi Fujiwara, and Daniel Vogel. 2021. Phonetroller: Visual Representations of Fingers for Precise Touch Input with Mobile Phones in VR. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 129, 13 pages. doi:10.1145/3411764.3445583

[17] Manuel Meier, Paul Streli, Andreas Fender, and Christian Holz. 2021. TapID: Rapid Touch Interaction in Virtual Reality using Wearable Sensing. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. 519–528. doi:10.1109/VR50410.2021.00076

[18] Takehiro Niikura, Yoshihiro Watanabe, and Masatoshi Ishikawa. 2014. Anywhere surface touch: utilizing any surface as an input area. In *Proceedings of the 5th Augmented Human International Conference* (Kobe, Japan) *(AH '14)*. Association for Computing Machinery, New York, NY, USA, Article 39, 8 pages. doi:10.1145/2582051.2582090

[19] Ju Young Oh, Ji-Hyung Park, and Jung-Min Park. 2020. FingerTouch: Touch Interaction Using a Fingernail-Mounted Sensor on a Head-Mounted Display for Augmented Reality. *IEEE Access* 8 (2020), 101192–101208. doi:10.1109/ACCESS.2020.2997972

[20] Shijia Pan, Ceferino Gabriel Ramirez, Mostafa Mirshekari, Jonathon Fagert, Albert Jin Chung, Chih Chi Hu, John Paul Shen, Hae Young Noh, and Pei Zhang. 2017. SurfaceVibe: vibration-based tap & swipe tracking on ubiquitous surfaces. In *Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks* (Pittsburgh, Pennsylvania) *(IPSN '17)*. Association for Computing Machinery, New York, NY, USA, 197–208. doi:10.1145/3055031.3055077

[21] Jean Peradel. 2017. Magic Frame : Turn Everything into a Touch Area. https://hackaday.io/project/27155-magic-frame-turn-everything-into-a-touch-area. [Accessed 29-August-2024].

[22] Vaninirappuputhenpurayil Gopalan Reju, Andy W. H. Khong, and Amir Bin Sulaiman. 2013. Localization of Taps on Solid Surfaces for Human-Computer Touch Interfaces. *IEEE Transactions on Multimedia* 15, 6 (2013), 1365–1376. doi:10.1109/TMM.2013.2264656

[23] Mark Richardson, Fadi Botros, Yangyang Shi, Pinhao Guo, Bradford J Snow, Linguang Zhang, Jingming Dong, Keith Vertanen, Shugao Ma, and Robert Wang. 2024. StegoType: Surface Typing from Egocentric Cameras. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) *(UIST '24)*. Association for Computing Machinery, New York, NY, USA, Article 83, 14 pages. doi:10.1145/3654777.3676343

[24] Mark Richardson, Matt Durasoff, and Robert Wang. 2020. Decoding Surface Touch Typing from Hand-Tracking. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 686–696. doi:10.1145/3379337.3415816

[25] Daniel Schneider, Verena Biener, Alexander Otte, Travis Gesslein, Philipp Gagel, Cuauhtli Campos, Klen Čopič Pucihar, Matjazz Kljun, Eyal Ofek, Michel Pahud, Per Ola Kristensson, and Jens Grubert. 2021. Accuracy Evaluation of Touch Tasks in Commodity Virtual and Augmented Reality Head-Mounted Displays. In *Proceedings of the 2021 ACM Symposium on Spatial User Interaction* (Virtual Event, USA) *(SUI '21)*. Association for Computing Machinery, New York, NY, USA, Article 7, 11 pages. doi:10.1145/3485279.3485283

[26] Changrui Shi, Ye Tao, Xiao Li, Shixin Li, Kaihao Mao, Wenshang Guo, Jian Zhou, Xiao Zhang, Rui Xue, and Yukun Ren. 2024. Grid-free touch recognition on arbitrary surface using triboelectric vibration sensor. *Nano Energy* 123 (2024), 109419. doi:10.1016/j.nanoen.2024.109419

[27] Yilei Shi, Haimo Zhang, Jiashuo Cao, and Suranga Nanayakkara. 2020. VersaTouch: A Versatile Plug-and-Play System that Enables Touch Interactions on Everyday Passive Surfaces. In *Proceedings of the Augmented Humans International Conference* (Kaiserslautern, Germany) *(AHs '20)*. Association for Computing Machinery, New York, NY, USA, Article 26, 12 pages. doi:10.1145/3384657.3384778

[28] Paul Streli, Jiaxi Jiang, Andreas Rene Fender, Manuel Meier, Hugo Romat, and Christian Holz. 2022. TapType: Ten-finger text entry on everyday surfaces via Bayesian inference. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 497, 16 pages. doi:10.1145/3491102.3501878

[29] Paul Streli, Jiaxi Jiang, Juliete Rossie, and Christian Holz. 2023. Structured Light Speckle: Joint Ego-Centric Depth Estimation and Low-Latency Contact Detection via Remote Vibrometry. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) *(UIST '23)*. Association for Computing Machinery, New York, NY, USA, Article 26, 12 pages. doi:10.1145/3586183.3606749

[30] Paul Streli, Mark Richardson, Fadi Botros, Shugao Ma, Robert Wang, and Christian Holz. 2024. TouchInsight: Uncertainty-aware Rapid Touch and Text Input for Mixed Reality from Egocentric Vision. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) *(UIST '24)*. Association for Computing Machinery, New York, NY, USA, Article 7, 16 pages. doi:10.1145/3654777.3676330

[31] Mayuka Tsuji, Hiroyuki Kubo, Suren Jayasuriya, Takuya Funatomi, and Yasuhiro Mukaigawa. 2021. Touch Sensing for a Projected Screen Using Slope Disparity Gating. *IEEE Access* 9 (2021), 106005–106013.

[32] Andrew D. Wilson. 2010. Using a depth camera as a touch sensor. In *ACM International Conference on Interactive Tabletops and Surfaces* (Saarbrücken, Germany) *(ITS '10)*. Association for Computing Machinery, New York, NY, USA, 69–72. doi:10.1145/1936652.1936665

[33] Robert Xiao, Scott Hudson, and Chris Harrison. 2016. DIRECT: Making Touch Tracking on Ordinary Surfaces Practical with Hybrid Depth-Infrared Sensing. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces* (Niagara Falls, Ontario, Canada) *(ISS '16)*. Association for Computing Machinery, New York, NY, USA, 85–94. doi:10.1145/2992154.2992173

[34] Robert Xiao, Julia Schwarz, Nick Throm, Andrew D. Wilson, and Hrvoje Benko. 2018. MRTouch: Adding Touch Input to Head-Mounted Mixed Reality. *IEEE Transactions on Visualization and Computer Graphics* 24, 4 (2018), 1653–1660. doi:10.1109/TVCG.2018.2794222

[35] Xing-Dong Yang, Tovi Grossman, Daniel Wigdor, and George Fitzmaurice. 2012. Magic finger: always-available input through finger instrumentation. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (Cambridge, Massachusetts, USA) *(UIST '12)*. Association for Computing Machinery, New York, NY, USA, 147–156. doi:10.1145/2380116.2380137

[36] Xing-Dong Yang, Khalad Hasan, Neil Bruce, and Pourang Irani. 2013. Surround-see: enabling peripheral vision on smartphones during active use. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology* (St. Andrews, Scotland, United Kingdom) *(UIST '13)*. Association for Computing Machinery, New York, NY, USA, 291–300. doi:10.1145/2501988.2502049

[37] Chun Yu, Xiaoying Wei, Shubh Vachher, Yue Qin, Chen Liang, Yueting Weng, Yizheng Gu, and Yuanchun Shi. 2019. HandSee: Enabling Full Hand Interaction on Smartphone with Front Camera-based Stereo Vision. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3290605.3300935

[38] Mingrui Ray Zhang and Jacob O. Wobbrock. 2019. Beyond the Input Stream: Making Text Entry Evaluations More Flexible with Transcription Sequences. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 831–842. doi:10.1145/3332165.3347922

[39] F. Zhu, Z. Lyu, M. Sousa, and T. Grossman. 2022. Touching The Droid: Understanding and Improving Touch Precision With Mobile Devices in Virtual Reality. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE Computer Society, Los Alamitos, CA, USA, 807–816. doi:10.1109/ISMAR55827.2022.00099