Above-Screen Fingertip Tracking with a Phone in Virtual Reality

Fabrice Matulic
fmatulic@preferred.jp
Preferred Networks
Japan

Taiga Kashima
kashima@nlab.ci.i.u-tokyo.ac.jp
Preferred Networks
Japan

Deniz Beker
denizbekerjp@gmail.com
Preferred Networks
Japan

Daichi Suzuo
suzuo@preferred.jp
Preferred Networks
Japan

Hiroshi Fujiwara
hfujiwara@preferred.jp
Preferred Networks
Japan

Daniel Vogel
dvogel@uwaterloo.ca
University of Waterloo
Canada

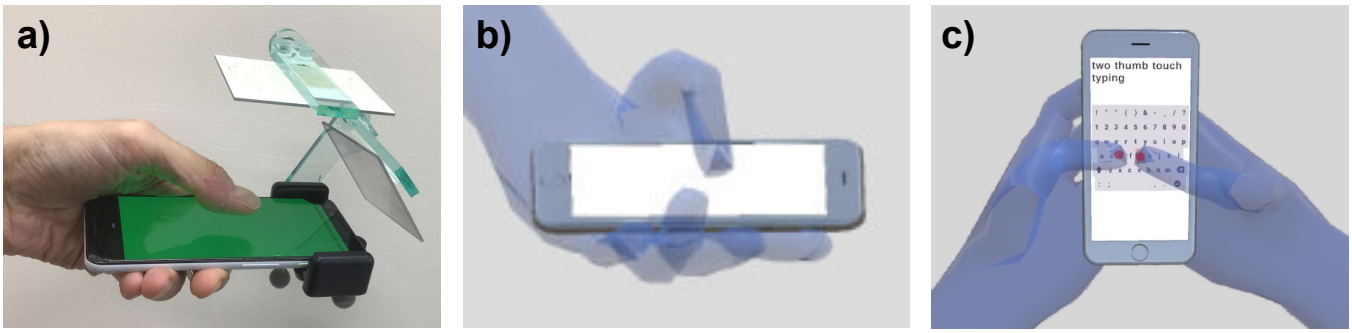

Figure 1: Phone-based fingertip-tracking system and hand models in VR: a) a custom mount with two mirrors captures the hand operating a phone from two different angles, 3D fingertip positions estimated with deep learning are used to control virtual hand hand models in VR (b) and c)).

Abstract

Using a phone as a VR controller is challenging because users cannot see their fingers when aiming for targets on the touchscreen. We propose using two mirrors mounted above the screen that reflect the front camera and a purpose-built deep neural network to robustly infer the 3D position of fingertips manipulating the phone. Network training is self-supervised after only a few initial labelled images and does not require any external sensor. We present a few example scenarios showing potential applications that use our phone-based fingertip tracker for precise touch input and above-screen interaction.

CCS Concepts

• **Human-centered computing** → **Interaction tech**.

Keywords

virtual reality, hand pose estimation

ACM Reference Format:

Fabrice Matulic, Taiga Kashima, Deniz Beker, Daichi Suzuo, Hiroshi Fujiwara, and Daniel Vogel. 2023. Above-Screen Fingertip Tracking with a Phone in Virtual Reality. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems (CHI EA '23), April 23–28, 2023, Hamburg, Germany*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3544549.3585728

CHI EA '23, April 23–28, 2023, Hamburg, Germany

ACM ISBN 978-1-4503-9422-2/23/04.
https://doi.org/10.1145/3544549.3585728

1 Introduction

Mobile phones are ubiquitous handheld touch devices that can also be used as a type of controller in virtual reality (VR) [5, 9, 21, 35]. A key challenge is how to render accurate feedback of users' hands and fingers as they operate the phone in VR. In particular, common touch input tasks, such as typing and tapping buttons, become difficult because a user wearing an immersive headset cannot see their fingers as they aim for targets on the touchscreen. Some headset sensors can track the hands and represent them as generic 3D models in the VR world, but current commercial systems cannot robustly determine poses of hands holding objects [15, 26] so they cannot be relied upon to precisely track hands and fingers operating a phone. One workaround is to directly show a camera feed of the segmented hand [3, 24], but the result is often noisy with visual artefacts that do not blend well with smooth computer-generated 3D VR scenes. Furthermore, hands moving outside of the camera's view cannot be captured. Another approach exploits specialised capacitive touchscreens to track finger hover [19], but phones with such capabilities are rare and currently not sold as new. Moreover, this style of hover tracking is imprecise [24] with feedback essentially limited to a 2D cursor for a single hovering fingertip.

We propose a technique that uses the front camera of a standard mobile phone to robustly track thumb and index fingertips above

the touchscreen. Our approach is inspired by MirrorTablet [20] and Phonetroller [24], both of which capture images of the hand via the front camera of a mobile device reflected by a mirror mounted above its screen. Instead of using a single mirror, we use a mount to place *two* mirrors at different angles in order to capture the hand from two distinct perspectives (Figure 1a). The image pairs are processed by a deep neural network to estimate the position of the fingertips in local 3D space. The network requires only a few labelled images initially and is otherwise trained in a self-supervised manner. Importantly, it does not require any external sensor to create labelled data thanks to differentiable rendering employed by our data-creation process. We present preliminary results of the tracking accuracy of our technique, followed by a few examples of applications and touch+above screen interactions enabled by it to inspire deeper investigations of that design space.

2 Related Work

2.1 Around-Phone Interaction Outside of VR

Several around-device hand and finger tracking techniques have been proposed to extend touch interaction with a mobile phone to mid-air gestures in non-VR contexts. Finger detection is performed via a camera [10, 38], a dedicated hand-tracking sensor [30], or capacitive proximity sensing [12, 16]. Some works focus on adding or extending the viewing range of a phone camera using mirrors reflecting the phone's cameras[43, 44]. Those systems either require an additional sensor, do not track hands above the screen, or are designed for coarse gesturing and thus are not suitable for precise 3D tracking of fingertips. HandSee places a prism with a mirror on the front camera to create stereoscopic images through double reflection for the detection of finger gestures above the screen [45]. No measured depth errors for fingertip tracking are reported, but the proximity of the two viewpoints of the two virtual cameras created by the prism likely makes it difficult to detect small movements in the depth dimension with high precision. Coverage is also limited in the upper area of the phone screen near the prism, especially for cameras with narrow fields of view.

Other sensors have also been used to detect hand gestures around mobile phones, e.g. magnets [27], millimetre-wave radar [22], GSM [48] and acoustic signals [29, 42, 46]. However, those solutions either require extra sensors or are sensitive to environmental noise.

2.2 Smartphones as VR Controllers

Smartphones have been considered for use as VR controllers, initially without visual feedback of the hand in the VR scene. User interfaces in those cases rely on coarse actions such as directional dragging and number of fingers touching the screen to operate the phone "blindly" [5, 9, 21, 35].

When visual feedback of the hand or finger(s) operating the phone is available, more classic user interfaces are possible, as users can aim more precisely for targets. Son et al [37] and HoVR-Type [19] track thumbs above the screen to aid typing on a mobile phone using respectively a motion capture system and the hover detection feature of a Samsung Galaxy S4. Those solutions are not practical for most VR contexts as they rely on costly or discontinued sensing hardware. Bai et al develop an augmented virtuality system that captures hands operating a phone with a depth camera mounted on the headset and renders them over a virtual phone in VR, which is aligned with the real device [3]. The rendering exhibits several artefacts due to the imperfect colour-based segmentation of the skin, the fingertips are not tracked and the camera on the headset must directly face the hands without occlusion to be able to capture them.

Phonetroller uses the front camera of the phone to capture a 2D image of hands manipulating the device through the reflection of a downward-facing mirror mounted above the screen [24]. Only the portion of the hand that is directly above the screen is shown as a 2D texture overlay in VR, which provides limited visual feedback with a greatly reduced sense of depth. Our system expands the idea of using a top mirror to capture the hand by adding a second mirror with a different orientation. With that second view, the 3D position of the fingertip can be inferred via deep learning and used to control various 3D hand models and objects, which harmoniously blend with the VR scene.

2.3 Hand-Object Pose Estimation

Current VR headsets and hand trackers can track hands with relatively high spatial accuracy, but even the best systems exhibit an average positional error of more than 1cm for fingertip detection [1, 36], which is twice the width of a key on a typical phone keyboard [14, 34]. Furthermore, these hand trackers are mainly designed for bare hands, not for hand-object interactions and thus have trouble robustly estimating hand poses in those situations [15, 26]. In TabletInVR, which investigates the use of a tablet device for modelling in VR [39], both hands are captured by a Leap Motion on the headset and materialised in the VR world, but the authors note that the sensor is sensitive to screen reflections and certain hand angles, which leads them to design their gestures around those constraints.

Computer vision techniques and datasets have been proposed to estimate the pose of hands holding and manipulating objects without markers [6, 40, 41], but they rely on cameras on the headset or in the environment and so tracking performance may degrade if the hands move to suboptimal viewing angles and distances. Furthermore, these techniques have not been applied to the specific scenario of precisely tracking a hand manipulating a mobile phone.

3 Self-Supervised 3D Fingertip Detection

Our goal is to track the user's fingertip(s) moving above and interacting with the phone screen in the local 3D space of the device with high precision using only the front camera as sensor. Knowing the phone's position (tracked by other means beyond the scope of this work), we can then obtain the fingertips' position in world space to control virtual hand models or other objects.

3.1 Two-Mirror System

For high tracking precision and coverage on all three dimensions using RGB images as input, multi-view capturing is preferable [8]. A common multi-view approach is to use stereo vision, but systems with a small baseline (distance between the two cameras) and similar capturing angles such as the prism used in HandSee [45] require perfect calibration and stereo matching to yield a low depth error. This is difficult to achieve with self-built equipment.

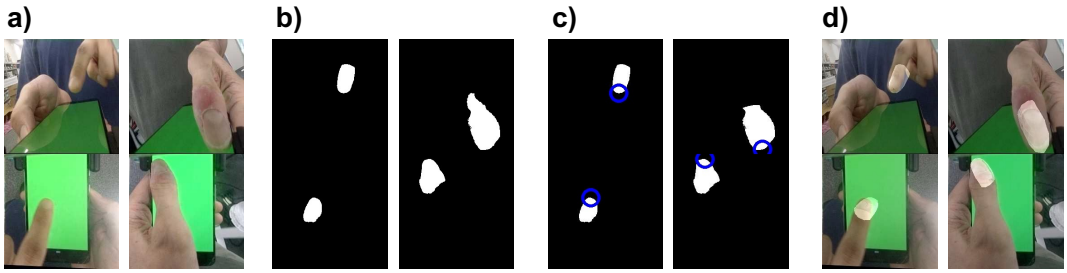

Figure 2: Data creation pipeline: a) data collection for thumbs and index fingers from both hands, b) extraction of mask regions of the fingers' distal phalanges, c) 2D fingertip location on the mask contours (blue circles) and mask refinement, d) differentiable rendering using the masks and the 2D fingertip position to obtain the 3D position and orientation of the phalanges.

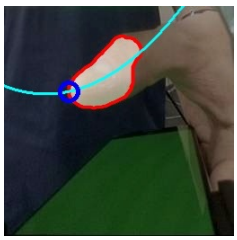

Figure 3: Polynomial curve intersecting the mask contour to determine the fingertip.

We therefore use a two-view setup with two mirrors mounted on the phone that reflect the front camera at two different positions and angles: One near-vertical mirror placed close to the camera, which produces a rear view, and a second mirror placed parallel to the screen above it, which produces a top view (Figure 1a). Both mirrors are positioned so that they each reflect the whole phone screen and the immediate space above it on half of the frame's vertical pixel space (Figure 2a).

3.1.1 Frame Preprocessing In a preprocessing step, the two (manually defined) regions corresponding to the two mirror views are cropped from the source camera frame and transformed into two square images. These images form the input of the deep learning pipeline that estimates the 3D pose of the tips of visible thumbs and index fingers.

3.1.2 Camera Calibration We treat each mirror's view as a virtual camera and perform standard calibration procedures using a mini ChArUco board and ArUco markers displayed on the phone's screen to obtain each camera's intrinsic and extrinsic parameters. Those parameters allow us to transform pixel coordinates into 3D points in the phone's coordinate system, whose origin we set to be the middle of the top bezel of the device.

3.2 Data Creation

To train a neural network that estimates the 3D position of fingertips based on RGB images, ground truth data with images of fingers labelled with the corresponding 3D positions of their tips is required. These labels can be automatically determined using an external sensor, such as a precise depth camera or a motion capture system that detects small optical markers. However, such equipment is typically expensive or cumbersome to set up. We propose a data creation method that does not rely on any external sensor and only requires training images obtained from people manipulating the phone.

The data creation pipeline consists of three steps, which are illustrated in Figure 2b, c and d: 1) Segmentation of the distal phalanx in the pair of input images, 2) detecting the 2D fingertip on the segmentation mask, and 3) using differentiable rendering to obtain the best-fitting 3D pose based on the mask silhouette and the 2D fingertip. We work only with distal phalanges and fingertips instead of entire hands or fingers, as it allows us to optimise for a rigid mesh, which is simpler than trying to fit a complex deformable 3D model.

3.2.1 Mask Estimation The first step of our pipeline is to determine a segmentation mask of the object to fit, i.e. the distal phalanx of the thumb or index finger that appears in each of the two input images. For robust segmentation we seek to use a CNN-based binary object segmentation neural network. Since there are no public datasets or neural network models optimised for the segmentation of distal phalanges, we use an existing pre-trained salient object detector, BASNet [31], which we fine-tune using a few manually labelled images (i.e. the boundaries of the phalanges are drawn by a human annotator). This step yields initial segmentation masks as shown in Figure 2b.

3.2.2 Fingertip Detection and Mask Refinement In this step we detect the location of the fingertip on the contour of the segmentation mask. This has two purposes: 1) It allows us to refine the mask by removing extra pixels beyond the distal interphalangeal (DIP) joint, which do not belong to the distal phalanx. 2) It adds a constraint for the neural renderer in the next step to match the 2D and 3D fingertip locations when optimising the position and orientation of the phalanx mesh.

To estimate the fingertip location on the mask contour, we first fit a polynomial curve to the mask pixels using regression, as we found a polynomial provides a good approximation of bent fingers when the distal phalanx is not cleanly segmented around the DIP joint. Depending on which of the image's x-y axes is used as the base axis, there are two possible curves that can be fit to the pixels. The correct axis is the one that is orthogonal to the rough orientation of the phalanx (residuals are calculated on the orthogonal axis), which can be determined from the aspect ratio of the mask's bounding box. The curve intersects the mask contour at least at two points, one of which is the approximate fingertip position. We determine the correct point, based on the knowledge of which hand is visible. For instance, for the rear view, if the digit is the left thumb, we pick the leftmost intersection point (Figure 3). For the top view, we choose the top point.

After we locate the fingertip on the contour, we refine the phalanx mask by performing a Boolean AND operation on the current mask with a disc mask centred on the fingertip and whose radius is experimentally determined so that pixels beyond the DIP joint are eliminated (Figure 2c).

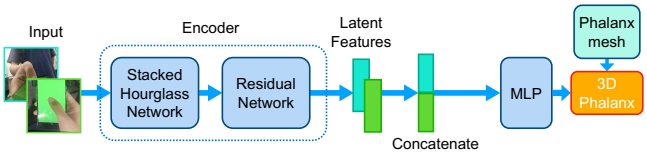

Figure 4: Adaptation of Ge et al's hand pose estimation network architecture [13] to handle input from multi-view image pairs. Each view is fed individually to the encoder. The two output latent feature vectors are concatenated in a single vector. This vector is, in turn, fed to the MLP branch that predicts the 3D fingertip pose and finger presence probability.

3.3 3D Pose And Location Estimation

The final step of the data-creation pipeline is the estimation of the 3D pose of the distal phalanx. To achieve this, we use differentiable mesh rendering, which is a self-supervised computer vision technique that can find correspondences between 3D objects and their appearances in 2D images [17, 18]. In our case, a differentiable renderer optimises the 6-DoF position and orientation of a 3D phalanx mesh so that its rendered form best overlaps with the 2D mask. For the mesh, we use the MANO hand model [33], from which we extract the vertices and faces belonging to the distal phalanges of the thumb and the index finger. Using this rigid, non-articulated subset of the finger meshes allows us to optimise only for 6-DoF position and orientation rather than a complex articulated hand pose. While people's finger sizes and shape differ, this model provides a sufficient approximation for our purpose.

For a considered phalanx mesh, rotation and translation matrices that orient and position the mesh are determined through an optimisation process, which iteratively minimises a loss function through gradient descent. At each step, the mesh is rendered for each virtual camera using Soft-Rasterizer [23] and the corresponding loss is calculated. This loss function is composed of three constituent losses: 1) a Silhouette Loss, which rotates and translates the 3D mesh towards the mask location, a Distribution Alignment Loss, which attempts to align the mesh silhouette and mask distributions for faster convergence and 3) a Tip Distance Loss which tries to align the projected 3D fingertip with the mask's 2D fingertip. (See Appendix for mathematical definitions of those losses.) Figure 2d shows an example of this optimisation process, which results in the 3D phalanx mesh aligned with the corresponding phalanx pixels in the input images.

The 3D pose of the phalanx is estimated for each frame separately, but we can expect the fingertip positions to remain close in consecutive frames. We make use of that fact by initialising the values for a new pose with the values estimated in the previous frame. This decreases the number of iterations required to arrive at an optimal solution.

3.4 Model training phase

While the data-creation process gives us an estimate of the 6-DoF pose of fingertips, it is typically slow and therefore not suitable for real-time inference, which is what we require for tracking. We therefore use the 3D position and orientation of fingertips obtained in the data-creation process as ground truth to train a dedicated end-to-end neural network, which is used for real-time inference. The goal for such a model is to directly output the pose estimate of the distal phalanx as well as a probability value indicating to which finger it belongs or "no finger" if no interacting fingers appear in the image (fingers used to hold the phone do not count as "interacting" fingers).

For our neural network base, we use Ge et al's graph CNN-based model [13], which is a popular architecture for 3D hand pose estimation. Since that network expects single-view RGB images as input, whereas we have image pairs from different views, we perform the forward pass of the pre-trained encoder with each image individually and concatenate the latent features obtained as output. We then feed the concatenated feature maps to the multi-layer perceptron (MLP) branch of the network to obtain the desired 3D pose and probabilities (Figure 4).

3.4.1 Handling multiple fingers Although the data-creation pipeline described above is designed to predict the pose of a single finger, it can be easily extended to handle multiple fingers without manually collecting additional data using mixup data augmentation [47]. Specifically, a source image is randomly chosen from the dataset, then a check is performed to determine which finger is shown (if any), and a complementary finger image is randomly picked (i.e. an image of a thumb or index finger from the other hand) to create a pair of images. If no finger is present in the source image, an image that includes any finger is randomly chosen. The two selected images are then blended to form a single combined image.

Models trained in such a way can be deployed for two-hand input, such as two-thumb typing (Figure 1c).

4 Enabling System

We create a proof-of-concept implementation of our technique using the following hardware and software components:

Mobile Phone For the phone, we use a Google Pixel 3. We create a mount for the two mirrors with an articulated arm made of acrylic glass attached to a holder clipped to the phone (Figure 1a). We currently use two flat mirrors, but smaller curved mirrors providing similar coverage could also possibly be used to slightly decrease weight. We attach optical markers to the mount to track the phone with an Optitrack motion capture system. We create a custom Android application that crops and transforms the two mirror images as described above and sends them along with touch data via WiFi to the inference server.

Neural Network Implementation We collect training data for our neural network from 11 people performing finger movements on and above the phone. We feed the data to our auto-labelling pipeline, where we manually annotate 350 images for each digit to improve BASNet segmentation. We collect $\sim 30,000$ images for the left hand and $\sim 35,000$ images for the right hand, with equal amounts of images for the thumb and index fingers, and $\sim 6,500$ "no finger" images for each hand. Our neural network pipeline is implemented in PyTorch with PyTorch3D [32] used for differentiable rendering. The models are trained for 25 epochs using the Adam optimiser.

Inference Server and VR Client The inference engine runs on a server with a GeForce RTX 3090 GPU, which achieves an inference time of $\sim 5ms$ per frame on average. After applying 1 euro filters [7] to the

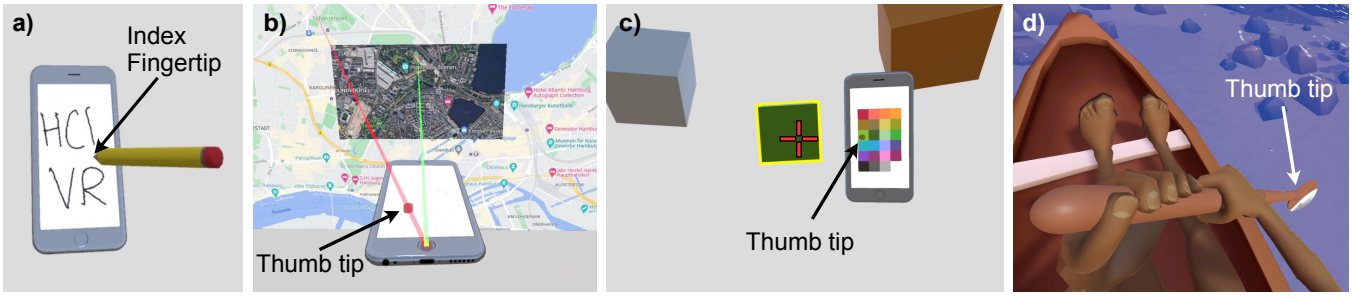

Figure 5: Applications enabled by our fingertip tracker: a) Sketching application with the fingertip represented as a pen; b) Double raycasting with phone + fingertip to create a rectangular viewing filter lens for map exploration; c) HUD interface with fixed phone position to assign texture colours chosen with touch to objects selected with head orientation; d) Canoeing game example, where the thumbtip controls the oar of a character in a canoe. Dragging on the phone screen rows the canoe

estimated fingertip position for smoothing, the data is sent to the VR client, which, for technical reasons, is deployed on a separate computer. We use a Vive Pro as VR system and the VR applications are developed in Unity.

The end-to-end latency of the whole processing chain measured by a high-speed camera is $\sim 125ms$.

5 Preliminary Performance Evaluation

To obtain an initial assessment of the tracking performance of our model, we compare the positions of the fingertip estimated by our technique with the 3D positions of an optical marker affixed to the top of the nail, as captured by the OptiTrack system. We recruit six people, who did not contribute data for training, and ask them to perform slow tapping motions on the whole screen successively with their thumb and index finger for three minutes. We compute the root mean squared error for each participant and each finger and obtain an average RMSE of $\sim 7mm$ for the thumb and $\sim 9mm$ for the index finger. These results are encouraging as our fingertip tracker for hands holding and interacting with phones demonstrates greater precision than barehand trackers of commercial HMDs. [1, 36].

6 Applications

With the estimated 3D fingertip position, hand models can be controlled for single-finger interaction to support common touch-based phone tasks in VR. Additionally, precise finger tracking enables a range of above-screen and touch-to-air techniques, a space explored by prior work in non-VR contexts [10, 16, 45] and in extended reality with tabletops, where hands can interact with virtual objects appearing above the device [11, 28, 39, 49]. We briefly present examples of applications and interactions leveraging our mobile fingertip tracker that go beyond simply replicating physical phone manipulations in VR. We leave the deeper investigation and evaluation of these techniques and the exploration of the underlying design space to future work.

6.1 Object Control with Finger-Differentiated Actions

Instead of controlling a hand model, the fingertip can be mapped to an object that more closely represents an input instrument for a particular task, similar to how VR controllers can transform into different handheld tools in VR applications (e.g. a pistol or a sword in a game). Task-specific representations can be considered at the finger(tip) level as well. For instance, in a sketching application, the index fingertip can control the 3D model of a pen, which virtually inks the screen of the phone (Figure 5a). Pen control and action can be swapped based on the used finger. For example, when using the thumb instead of the index finger, the pen could flip to its eraser end and dragging the thumb on the screen would then erase content.

6.2 Double Raycasting

The tracked phone can be used as a 6-DoF raycasting source in the virtual 3D space like a standard VR controller [2, 24, 35]. Since the fingertip is also precisely tracked in the local 3D space of the phone, it can be used as an anchor for a second ray to support double raycasting with a single hand, similar to barehand pointing techniques using multiple fingers for remote interaction with large displays [4, 25]. Our example in VR considers a resizable rectangular lens or filter to explore maps, such as revealing the satellite image of a particular portion of the map (Figure 5b). The centre of the lens is determined by the phone ray, and the width and height by a second ray emanating from the centre of the bottom bezel of the phone and passing through the fingertip. If needed, an action (e.g. a selection confirmation) can be triggered by pressing the physical phone's volume button.

6.3 Use as Head-Up Display

Head-up interfaces, which have a fixed position in the 2D screen space of the viewport, are often used in XR to show information to the user. Head-up displays or HUDs can also support phone-operated menus [24]. In such a setting, the phone does not need to be tracked as it is used only for its touch capabilities (with 3D visual feedback of the hand to enable precise targeting). Since fingers are tracked inside-out via the phone itself and not by the HMD, the user is free to look in any direction without impacting fingertip detection accuracy. This also allows the phone to be held in a comfortable, low-fatigue position, such as resting it on the lap or a table when sitting. This scenario cannot be supported by existing VR systems with hand-tracking sensors integrated in the HMD, which expect the hands to be in view (i.e. directly in front of the HMD) to capture them. We present an application of this HUD concept, where head

orientation or gaze is used to point at 3D objects located around the user in the VR space and the phone is used to choose texture colours from a palette for selected objects (Figure 5c).

6.4 Touch/Drag Navigation + Mid-Air Interaction

Scenarios in which the representation of and interaction with the physical phone are significantly abstracted in the VR world can also be considered. Expanding on the idea of using both touch and above-screen input, we propose a novel interaction paradigm for games, in which the tracked fingertip is mapped to the control point of a character and dragging on the phone screen moves the character. One game scenario in which dragging on a surface is a strong metaphor for motion control is rowing. The fingertip is mapped to the virtual oar of a character sitting in a canoe (Figure 5d) and dragging the finger on the screen causes the character to row the canoe. The dragging length, speed and direction control the speed and direction of the boat via the virtual rowing actions.

Mid-air interaction can also be an integral part of the game, for example, the rower can raise their oar to try to hit overhead targets when passing under them. The orientation of the phone as detected by its internal motion sensors can further be used to tilt the canoe for added control. Here again, the phone does not need to be tracked, making such an application easily deployable on inexpensive mobile 3-DoF VR systems which do not track any devices or hands. Furthermore, we speculate that single-finger control of a character with a phone that can be held in a relaxed pose is likely significantly less fatiguing than using a standard VR controller, whose movements are mapped 1-to-1 to physical leg and arm motion. Dragging with a finger on a phone screen to row is an inexpensive but serviceable approximation of haptic feedback and friction when an oar pushes water backwards.

7 Conclusion

We proposed a deep learning technique to track the 3D position of thumb and index fingertips on a mobile phone in VR using images captured by the front camera reflected through two mirrors mounted on the device. Our method requires no external sensor for training or inference and only a few images need to be initially labelled, with ground truth data mainly generated in a self-supervised manner through differentiable rendering. We reported initial results of tracking accuracy and presented several applications uniquely enabled by our technique. We plan to investigate the potential for precise finger interaction with mobile phones in VR more deeply in future work.

Acknowledgments

We thank Sosuke Kobayashi for generously donating his Pixel 3 phone for this project.

References

[1] Diar Abdlkarim, Massimiliano Di Luca, Poppy Aves, Sang-Hoon Yeo, R Chris Miall, Peter Holland, and Joseph M Galea. 2022. A Methodological Framework to Assess the Accuracy of Virtual Reality Hand-Tracking Systems: A case study with the Oculus Quest 2. *bioRxiv* (2022).

[2] Teo Babic, Harald Reiterer, and Michael Haller. 2018. Pocket6: A 6DoF Controller Based On A Simple Smartphone Application. In *Proceedings of the Symposium on Spatial User Interaction* (Berlin, Germany) *(SUI '18)*. Association for Computing Machinery, New York, NY, USA, 2–10. https://doi.org/10.1145/3267782.3267785

[3] Huidong Bai, Li Zhang, Jing Yang, and Mark Billinghurst. 2021. Bringing full-featured mobile phone interaction into virtual reality. *Computers & Graphics* 97 (2021), 42–53. https://doi.org/10.1016/j.cag.2021.04.004

[4] Amartya Banerjee, Jesse Burstyn, Audrey Girouard, and Roel Vertegaal. 2012. MultiPoint: Comparing Laser and Manual Pointing as Remote Input in Large Display Interactions. *Int. J. Hum.-Comput. Stud.* 70, 10 (oct 2012), 690–702. https://doi.org/10.1016/j.ijhcs.2012.05.009

[5] Sabah Boustila, Thomas Guégan, Kazuki Takashima, and Yoshifumi Kitamura. 2019. Text Typing in VR Using Smartphones Touchscreen and HMD. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, New York, NY, USA, 860–861. https://doi.org/10.1109/VR.2019.8798238

[6] Samarth Brahmbhatt, Chengcheng Tang, Christopher D Twigg, Charles C Kemp, and James Hays. 2020. ContactPose: A Dataset of Grasps with Object Contact and Hand Pose. In *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham, 361–378.

[7] Géry Casiez, Nicolas Roussel, and Daniel Vogel. 2012. 1 € Filter: A Simple Speed-Based Low-Pass Filter for Noisy Input in Interactive Systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Austin, Texas, USA) *(CHI '12)*. Association for Computing Machinery, New York, NY, USA, 2527–2530. https://doi.org/10.1145/2207676.2208639

[8] Liangjian Chen, Shih-Yao Lin, Yusheng Xie, Yen-Yu Lin, and Xiaohui Xie. 2021. MVHM: A Large-Scale Multi-View Hand Mesh Benchmark for Accurate 3D Hand Pose Estimation. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 836–845. https://doi.org/10.1109/WACV48630.2021.00088

[9] Sibo Chen, Junce Wang, Santiago Guerra, Neha Mittal, and Soravis Prakkamakul. 2019. Exploring Word-Gesture Text Entry Techniques in Virtual Reality. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI EA '19)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3290607.3312762

[10] Xiang 'Anthony' Chen, Julia Schwarz, Chris Harrison, Jennifer Mankoff, and Scott E. Hudson. 2014. Air+touch: Interweaving Touch & in-Air Gestures. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) *(UIST '14)*. Association for Computing Machinery, New York, NY, USA, 519–525. https://doi.org/10.1145/2642918.2647392

[11] Bruno R De Araújo, Géry Casiez, Joaquim A Jorge, and Martin Hachet. 2013. Mockup Builder: 3D modeling on and above the surface. *Computers & Graphics* 37, 3 (2013), 165–178. https://doi.org/10.1016/j.cag.2012.12.005

[12] Li Du, ChunChen Liu, Adrian Tang, Yan Zhang, Yilei Li, Kye Cheung, and Mau-Chung Frank Chang. 2016. Airtouch: A novel single layer 3D touch sensing system for human/mobile devices interactions. In *2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. 1–6. https://doi.org/10.1145/2897937.2901902

[13] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 2019. 3D Hand Shape and Pose Estimation from a Single RGB Image. In *CVPR*.

[14] Ewa Gustafsson, Pieter Coenen, Amity Campbell, and Leon Straker. 2018. Texting with touchscreen and keypad phones - A comparison of thumb kinematics, upper limb muscle activity, exertion, discomfort, and performance. *Applied Ergonomics* 70 (2018), 232–239. https://doi.org/10.1016/j.apergo.2018.03.003

[15] Shangchen Han, Beibei Liu, Randi Cabezas, Christopher D. Twigg, Peizhao Zhang, Jeff Petkau, Tsz-Ho Yu, Chun-Jung Tai, Muzaffer Akbay, Zheng Wang, Asaf Nitzan, Gang Dong, Yuting Ye, Lingling Tao, Chengde Wan, and Robert Wang. 2020. MEgATrack: Monochrome Egocentric Articulated Hand-Tracking for Virtual Reality. *ACM Trans. Graph.* 39, 4, Article 87 (jul 2020), 13 pages. https://doi.org/10.1145/3386569.3392452

[16] Ken Hinckley, Seongkook Heo, Michel Pahud, Christian Holz, Hrvoje Benko, Abigail Sellen, Richard Banks, Kenton O'Hara, Gavin Smyth, and William Buxton. 2016. Pre-Touch Sensing for Mobile Interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 2869–2881. https://doi.org/10.1145/2858036.2858095

[17] Hiroharu Kato, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. 2020. Differentiable Rendering: A Survey. *CoRR* abs/2006.12057 (2020). arXiv:2006.12057 https://arxiv.org/abs/2006.12057

[18] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. 2018. Neural 3D Mesh Renderer. In *CVPR*.

[19] Youngwon R. Kim and Gerard J. Kim. 2017. HoVR-Type: Smartphone as a typing interface in VR using hovering. In *2017 IEEE International Conference on Consumer Electronics (ICCE)*. 200–203. https://doi.org/10.1109/ICCE.2017.7889285

[20] Khanh-Duy Le, Kening Zhu, and Morten Fjeld. 2017. Mirrortablet: Exploring a Low-Cost Mobile System for Capturing Unmediated Hand Gestures in Remote Collaboration. In *Proceedings of the 16th International Conference on Mobile and Ubiquitous Multimedia* (Stuttgart, Germany) *(MUM '17)*. Association for Computing Machinery, New York, NY, USA, 79–89. https://doi.org/10.1145/3152832.3152838

[21] Hai-Ning Liang, Yuwei Shi, Feiyu Lu, Jizhou Yang, and Konstantinos Papangelis. 2016. VRMController: An Input Device for Navigation Activities in Virtual Reality Environments. In *Proceedings of the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry - Volume 1* (Zhuhai, China) *(VRCAI '16)*. Association for Computing Machinery, New York, NY, USA, 455–460. https://doi.org/10.1145/3013971.3014005

[22] Jaime Lien, Nicholas Gillian, M. Emre Karagozler, Patrick Amihood, Carsten Schwesig, Erik Olson, Hakim Raja, and Ivan Poupyrev. 2016. Soli: Ubiquitous Gesture Sensing with Millimeter Wave Radar. *ACM Trans. Graph.* 35, 4, Article 142 (July 2016), 19 pages. https://doi.org/10.1145/2897824.2925953

[23] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. 2019. Soft Rasterizer: A Differentiable Renderer for Image-Based 3D Reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[24] 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. https://doi.org/10.1145/3411764.3445583

[25] Fabrice Matulic and Daniel Vogel. 2018. Multiray: Multi-Finger Raycasting for Large Displays. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3173574.3173819

[26] Fabrice Matulic and Daniel Vogel. 2022. Terrain Modelling with a Pen & Touch Tablet and Mid-Air Gestures in Virtual Reality. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI EA '22)*. Association for Computing Machinery, New York, NY, USA, Article 301, 7 pages. https://doi.org/10.1145/3491101.3519867

[27] Jess McIntosh, Paul Strohmeier, Jarrod Knibbe, Sebastian Boring, and Kasper Hornbæk. 2019. Magnetips: Combining Fingertip Tracking and Haptic Feedback for Around-Device Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3290605.3300638

[28] Daniel Mendes, Fernando Fonseca, Bruno Araùjo, Alfredo Ferreira, and Joaquim Jorge. 2014. Mid-air interactions above stereoscopic interactive tables. In *2014 IEEE Symposium on 3D User Interfaces (3DUI)*. 3–10. https://doi.org/10.1109/3DUI.2014.6798833

[29] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. 2016. FingerIO: Using Active Sonar for Fine-Grained Finger Tracking. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 1515–1525. https://doi.org/10.1145/2858036.2858580

[30] Jing Qian, Jiaju Ma, Xiangyu Li, Benjamin Attal, Haoming Lai, James Tompkin, John F. Hughes, and Jeff Huang. 2019. Portal-Ble: Intuitive Free-Hand Manipulation in Unbounded Smartphone-Based Augmented Reality. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 133–145. https://doi.org/10.1145/3332165.3347904

[31] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. 2019. BASNet: Boundary-Aware Salient Object Detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[32] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. 2020. Accelerating 3D Deep Learning with PyTorch3D. *arXiv:2007.08501* (2020).

[33] Javier Romero, Dimitrios Tzionas, and Michael J. Black. 2017. Embodied Hands: Modeling and Capturing Hands and Bodies Together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)* 36, 6 (Nov. 2017).

[34] Eun Jeong Ryu, Minhyeok Kim, Joowoo Lee, Soomin Kim, Jiyoung Hong, Jieun Lee, Minhaeng Cho, and Jinhae Choi. 2016. Designing Smartphone Keyboard for Elderly Users. In *HCI International 2016 – Posters' Extended Abstracts*, Constantine Stephanidis (Ed.). Springer International Publishing, Cham, 439–444.

[35] Elaheh Samimi and Robert J. Teather. 2022. Multi-Touch Smartphone-Based Progressive Refinement VR Selection. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. 582–583. https://doi.org/10.1109/VRW55335.2022.00142

[36] 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 *Symposium on Spatial User Interaction* (Virtual Event, USA) *(SUI '21)*. Association for Computing Machinery, New York, NY, USA, Article 7, 11 pages. https://doi.org/10.1145/3485279.3485283

[37] Jeongmin Son, Sunggeun Ahn, Sunbum Kim, and Geehyuk Lee. 2019. Improving Two-Thumb Touchpad Typing in Virtual Reality. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI EA '19)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3290607.3312926

[38] Jie Song, Gábor Sörös, Fabrizio Pece, Sean Ryan Fanello, Shahram Izadi, Cem Keskin, and Otmar Hilliges. 2014. In-Air Gestures around Unmodified Mobile Devices. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (Honolulu, Hawaii, USA) *(UIST '14)*. Association for Computing Machinery, New York, NY, USA, 319–329. https://doi.org/10.1145/2642918.2647373

[39] Hemant Bhaskar Surale, Aakar Gupta, Mark Hancock, and Daniel Vogel. 2019. TabletInVR: Exploring the Design Space for Using a Multi-Touch Tablet in Virtual Reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300243

[40] Xiao Tang, Xiaowei Hu, Chi-Wing Fu, and Daniel Cohen-Or. 2020. GrabAR: Occlusion-Aware Grabbing Virtual Objects in AR. 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, 697–708. https://doi.org/10.1145/3379337.3415835

[41] Catherine Taylor, Murray Evans, Eleanor Crellin, Martin Parsons, and Darren Cosker. 2021. Ego-Interaction: Visual Hand-Object Pose Correction for VR Experiences. In *Motion, Interaction and Games* (Virtual Event, Switzerland) *(MIG '21)*. Association for Computing Machinery, New York, NY, USA, Article 1, 8 pages. https://doi.org/10.1145/3487983.3488290

[42] Wei Wang, Alex X. Liu, and Ke Sun. 2016. Device-Free Gesture Tracking Using Acoustic Signals. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking* (New York City, New York) *(MobiCom '16)*. Association for Computing Machinery, New York, NY, USA, 82–94. https://doi.org/10.1145/2973750.2973764

[43] Pui Chung Wong, Hongbo Fu, and Kening Zhu. 2016. Back-Mirror: Back-of-Device One-Handed Interaction on Smartphones. In *SIGGRAPH ASIA 2016 Mobile Graphics and Interactive Applications* (Macau) *(SA '16)*. Association for Computing Machinery, New York, NY, USA, Article 13, 2 pages. https://doi.org/10.1145/2999508.2999512

[44] 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. https://doi.org/10.1145/2501988.2502049

[45] 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* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300935

[46] Sangki Yun, Yi-Chao Chen, Huihuang Zheng, Lili Qiu, and Wenguang Mao. 2017. Strata: Fine-Grained Acoustic-Based Device-Free Tracking. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services* (Niagara Falls, New York, USA) *(MobiSys '17)*. Association for Computing Machinery, New York, NY, USA, 15–28. https://doi.org/10.1145/3081333.3081356

[47] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond Empirical Risk Minimization. *International Conference on Learning Representations* (2018). https://openreview.net/forum?id=r1Ddp1-Rb

[48] Chen Zhao, Ke-Yu Chen, Md Tanvir Islam Aumi, Shwetak Patel, and Matthew S. Reynolds. 2014. SideSwipe: Detecting in-Air Gestures around Mobile Devices Using Actual GSM Signal *(UIST '14)*. Association for Computing Machinery, New York, NY, USA, 527–534. https://doi.org/10.1145/2642918.2647380

[49] Fengyuan Zhu and Tovi Grossman. 2020. BISHARE: Exploring Bidirectional Interactions Between Smartphones and Head-Mounted Augmented Reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3313831.3376233