ShadowTouch: Enabling Free-Form Touch-Based Hand-to-Surface Interaction with Wrist-Mounted Illuminant by Shadow Projection

Chen Liang
liang-c19@mails.tsinghua.edu.cn
Tsinghua University
Beijing, China

Xutong Wang
wangxuto18@mails.tsinghua.edu.cn
Tsinghua University
Beijing, China

Zisu Li
zlihe@connect.ust.hk
The Hong Kong University of Science and Technology
Hong Kong SAR, China

Chi Hsia
xq22@mails.tsinghua.edu.cn
Tsinghua University
Beijing, China

Mingming Fan
mingmingfan@ust.hk
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
The Hong Kong University of Science and Technology
Hong Kong SAR, China

Chun Yu
chunyu@tsinghua.edu.cn
Tsinghua University
Beijing, China

Yuanchun Shi*
shiyc@tsinghua.edu.cn
Tsinghua University
Beijing, China
Qinghai University
Xining, China

ABSTRACT

We present ShadowTouch, a novel sensing method to recognize the subtle hand-to-surface touch state for independent fingers based on optical auxiliary. ShadowTouch mounts a forward-facing light source on the user's wrist to construct shadows on the surface in front of the fingers when the corresponding fingers are close to the surface. With such an optical design, the subtle vertical movements of near-surface fingers are magnified and turned to shadow features cast on the surface, which are recognizable for computer vision algorithms. To efficiently recognize the touch state of each finger, we devised a two-stage CNN-based algorithm that first extracted all the fingertip regions from each frame and then classified the touch state of each region from the cropped consecutive frames. Evaluations showed our touch state detection algorithm achieved a recognition accuracy of 99.1% and an F-1 score of 96.8% in the leave-one-out cross-user evaluation setting. We further outlined the hand-to-surface interaction space enabled by ShadowTouch's sensing capability from the aspects of touch-based interaction, stroke-based interaction, and out-of-surface information and developed four application prototypes to showcase ShadowTouch's interaction potential. The usability evaluation study showed the advantages of ShadowTouch over threshold-based techniques in aspects of lower mental demand, lower effort, lower frustration, more willing to use, easier to use, better integrity, and higher confidence.

CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous and mobile devices**; **Gestural input**.

KEYWORDS

touch detection, hand-to-surface interaction, computer vision

ACM Reference Format:
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 *The 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23), October 29–November 01, 2023, San Francisco, CA, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3586183.3606785

1 INTRODUCTION

Gestural interaction with bare hands is becoming an essential modality for the latest mixed reality (MR) interfaces [35]. Among the vast hand interaction space in mixed reality, touch interaction with a rigid physical surface serves a unique and significant role because such an input modality is most similar to the widely-adopted

*indicates the corresponding author.

UIST '23, October 29–November 01, 2023, San Francisco, CA, USA

ACM ISBN 979-8-4007-0132-0/23/10…$15.00
https://doi.org/10.1145/3586183.3606785

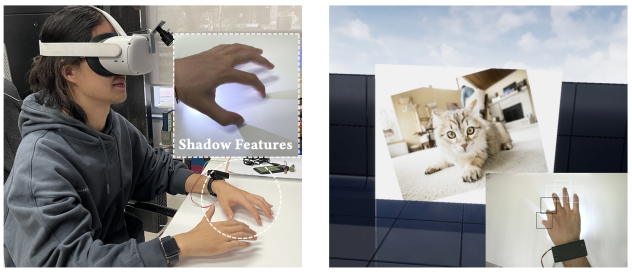

Figure 1: An example VR usage scenario enabled by ShadowTouch. (a) The user is sitting at a desk and browsing a photo album in VR environment. (b) With ShadowTouch, the user can turn the physical surface into a multi-touch interface, and perform a thumb-index multi-stroke gesture with aligned haptic feedback to make transformations of the photo.

touchscreen or touchpad interface and provides perfect tactile feedback [25] that helps to reduce fatigue [11, 25] and improve input efficiency [2, 3, 10, 26].

To enable rich hand interaction for mixed reality, vision-based hand tracking, as the fundamental sensing capability, has been extensively researched in the computer vision area. Although state-of-the-art commercial products (e.g., Microsoft Hololens 2 [16] and Oculus Quest 2 [18]) have demonstrated some potential of their hand tracking system in sensing pose-driven gestures, the sensing capability is still restricted in both spatial and temporal resolution [14], especially for the near-surface scenarios where the finger touches are fast and subtle. For example, a transient finger-to-surface touch event would hardly be distinguished from a false positive (e.g., pretending to touch) through a commercial camera system because a touch event typically happens within 100 milliseconds with a submillimeter spatial resolution (e.g., touch v.s. slight hovering) [46] so that the camera failed to distinguish the touch state between the finger and the surface.

Aiming at this challenge, previous work leveraged complementary channels, such as the inertial signal (captured by IMUs), to detect either touch events (one-finger [14] or multi-finger [35]) or touch states (one-finger [46]) by capturing the micro-vibration signal yielded from finger touches. Here we emphasize the difference between a touch event and a touch state - the former means a transient "clicking" event involving a touch-down and a touch-up phase while the latter means whether a certain finger contacts with the surface at a certain moment. Typically, detecting the touch state is known as a much harder task since it decouples the touch-down and touch-up phases [46], where touch-downs are more recognizable and touch-ups are implicit to the inertial signal.

In this paper, we aimed to recognize the subtle touch states of independent fingers, which is the fundamental sensing goal for hand-to-surface interaction, from a unique perspective by constructing an optical auxiliary. We presented ShadowTouch, a novel vision-based sensing technique to recognize subtle finger-to-surface touch states by actively constructing recognizable optical features to magnify the subtle finger movement near the surface. To achieve this goal, ShadowTouch mounts a forward-facing light source on the wrist to cast shadows of fingers onto the near surface, turning the subtle vertical movement of the fingers into amplified shadow features on the surface, as shown in Figure 1. Such an optical design is well complementary to vision-based hand tracking, reusing the camera channel to achieve subtle near-surface finger movement sensing.

The key of ShadowTouch is to construct high-quality and recognizable shadow features, so we first conducted an analysis on the optical principles of how the shadows were yielded and what characteristics good shadows should meet, outputting three main goals - sufficient magnification, recognizable contrast, and rich gesture space - for the target shadows. Then we justified our design considerations regarding the hardware form and hyperparameters to finalize our hardware design.

After acquiring recognizable shadow features, we devised a lightweight two-stage CNN-based model to recognize the touch states of independent fingers. Our model first extracted the boundaries of fingertip regions with dynamic sizes and then classified the cropped frame series within a short period to acquire the touch state for each finger. The evaluation showed our algorithm pipeline achieved an average accuracy of 99.0% and an F-1 score of 96.7% for touch state recognition using a 5-frame window in the cross-user setting.

To demonstrate ShadowTouch's applicability, we discuss the hand-to-surface interaction space enabled by ShadowTouch's sensing capability from the aspects of touch-based interaction, stroke-based interaction, and out-of-surface information. We also developed four application prototypes to showcase ShadowTouch's interaction space. The usability evaluation study showed ShadowTouch achieved a significantly higher resolution to distinguish finger touch states compared with collision-based algorithms, while applications enabled by ShadowTouch were well accepted by participants for lower mental demand, stronger willingness, higher easiness, higher integrity, and stronger confidence in usage compared with threshold-based techniques.

To sum up, we feature three main contributions of our work:

- We present the concept of ShadowTouch, a novel sensing scheme leveraging wrist-emitted light to construct recognizable shadow features for indicating the accurate touch state of near-surface fingers.
- We proposed a prototypical implementation of ShadowTouch, along with the hardware design and a lightweight two-stage model to recognize the near-surface touch states of independent fingers, achieving a recognition accuracy of 99.1% and an F-1 score of 96.8%.
- We demonstrated the rich hand-to-surface gesture space empowered by ShadowTouch and presented application prototypes to showcase ShadowTouch's interaction space. Our usability study validated the effectiveness of ShadowTouch compared with threshold-based techniques.

2 RELATED WORK

2.1 Physical-Aligned Touch Interaction in Immersive Environment

Adopting physical settings or surfaces in the environment to provide haptic feedback in VR/AR has found interest in Mixed Reality

communities. Offering passive haptic feedback with the help of physical entities in the environment has more advantages in reducing fatigue [11, 25], improving the accuracy of hand gesture input [25, 59] or sketch input[3, 25], enriching the set of interaction paradigms [19], etc. For instance, Yang et al. [66] adopted the user's skin as a convenient surface for tactile touch-driven interactions to enable precise on-skin touch segmentation. Also using the human body for tactile feedback, Fang et al. [11] explored self-haptics where the user's right hand physically feels a keypad surface for interaction. Wang et al. [51] and Liang et al. [29] investigated stroke-based interaction on the palm and the fingertip respectively to enable efficient interaction with good self-haptic feedback. OmniTouch [19] enabled the user to use their hands, arms, and legs as graphical, interactive surfaces as well as appropriate surfaces from the environment to expand the interactive area. For 3D sketch input tasks in VR/AR, touching a rigid physical surface for sketching has proved to improve the accuracy [2, 3, 10, 25, 26, 44]. Arora et al. [3] conducted a study that indicated drawing on a physical surface in VR performed better than projecting to virtual planes in the accuracy of the sketching. Jiang et al. [25] implemented VR physical gloves that enabled one user's non-dominant hand as a canvas and the dominant hand as the painter to provide physical feedback in 3D sketching.

Our work shared a similar goal with these works of bringing physical surfaces to hand interaction in MR to provide tactile feedback for better user experiences. Further, since the sensing scheme of ShadowTouch was irrelevant to the surface, it has more interactive capabilities for the ubiquitous hand input everywhere, not relying on any augmented surfaces or created overlays. We envisioned any flat physical surfaces in the environment to be interactable with ShadowTouch.

2.2 Recognizing Touch on Unmodified Surface

Researchers have been seeking solutions to sense touches on unmodified surfaces with various sensing methods, including IMU [14, 15, 30, 41], vibration sensing [21, 32, 46], pressure sensing [23, 52, 61], acoustic sensing [20, 27, 43, 56], and optical sensing [1, 12, 17, 36, 38].

The touch, based on the interaction context, could be explained as either the touch event [14, 15] or the touch state [35, 46]. The touch event refers to a transient "clicking" gesture with a touch-down phase immediately followed by a touch-up phase, while the touch state means whether a specific finger contacts the surface at a specific moment. Typically, detecting the touch state is known as a much harder task since it decouples the touch-down and touch-up phases [46], where touch-downs are more recognizable and touch-ups are implicit to the inertial signal. These two different sensing subjective (touch event v.s. touch state) rely on different sensing principles and could support different interaction spaces.

For sensing touch events, IMUs are most frequently used because they are suitable for capturing subtle movements and vibrations. For example, AnywhereTouch [41] proposed a finger-tracking method using nailed-mounted IMU on arbitrary surfaces. Gu et al. [14, 15] implemented a finger ring with IMU to detect the touch events of the index finger and further supported text entry on physical surfaces. TypeAnywhere [64] used a wearable device that straps individually for fingers to decode typing sequences based only on finger-tap sequences without relying on tap locations. TapID [34, 35] proposed a wrist-worn IMU device to sense precise touch events of an individual finger. Researchers also investigated acoustic methods [20, 27, 43, 56] to classify hand touching gestures, general hand-to-surface activities, and objects that actively emit acoustic signals [46]. For example, AudioTouch [27] enabled a system that requires attaching two piezo-electric elements, acting as a surface-mounted speaker and microphone, on the back of the hand to sense micro tap-based gestures.

Regarding the touch state, existing detection methods are largely based on certain sensor thresholds (e.g., magnetic field sensors [6], optical sensors [38, 57], and depth cameras [1, 35]). For instance, Magic Finger [57] integrated an optical mouse sensor and an RGB camera into a device worn on the fingertip. 3DTouch [38] coupled an optical laser sensor with a 9-DOF inertial measurement unit to support 3D interaction techniques, such as selection, translation, and rotation. Agarwal et al. [1] detected touch locations and moments of contact with overhead stereo cameras, discussing how the noise from stereo depth causes erroneous proximity readings. Optical sensing methods using depth cameras could help to recognize touch location, but the noisy depth data for reliable event detection is limited by the camera's depth resolution and frame rates [35]. Other than threshold-based methods, Shi et al. managed to use a finger-worn IMU to sense the touch state of a certain finger by distinguishing the vibration features between a supporting finger and a non-supporting finger [46].

As most related to our work, a number of research have further investigated the feasibility of recognizing finger touch states with enhanced vision-based methods by observing implicit optical features such as specular features [7, 60], shadow features[24, 45, 47, 48, 50], heat images [5, 28], and fingernail images [13, 49]. For example, HeatWave [28] leveraged thermal imaging cameras to detect finger touch, shape-based gestures, and pressure-based gestures on arbitrary surfaces, while ContactDB [5] analyzed and predicted grasp contact between hand and objects with thermal imaging. Local image changes in colors and textures on [13, 49] and around [13, 45] also served as essential features to indicate the contact state of the finger. Sekiya et al [45] presented a novel method to detect finger-to-skin contact by recognizing the shadows and texture around fingertips from skin deformation. PressureVision [13] presented an end-to-end model to estimate hand pressure map from a single RGB image.

In line with these works, ShadowTouch also focuses on the recognition of independent finger touch states with pure vision-based solutions. Different from existing vision-based methods, ShadowTouch actively constructed shadow features to amplify the subtle near-surface finger movements to achieve higher spatial resolution. Regarding the hardware form, ShadowTouch used a wrist-worn device that is not invasive to users' daily touching or hand activities compared with the finger-worn form [42].

2.3 Enhancing Vision-based Hand Gesture Recognition with Auxiliary Optical Instruments

Deploying auxiliary optical instruments to improve the capability of vision-based hand recognition has been extensively researched.

Constructing and utilizing reflection on mirror-like surfaces is a prevalent choice to enhance vision-based hand gesture sensing since it can provide multiple views of the observing hand and enable stereo gesture recognition [7, 31, 33, 55, 58, 62]. Matulic et al. [33] proposed a mounting mirror above the phone screen so that the front-facing camera captures the thumbs on or near the screen. Lim et al. [31] utilized plane mirrors to create multiple views of the hand to reduce the issues of occlusion and the error in measuring the flexion angle of finger joints. Yang et al. [58] attached an omni-directional mirror to the front camera to enable peripheral vision around the device. Yu et al. [62] used a prism mirror placed on the front camera to create a stereo vision system that generates a depth image of the hand for tracking finger location and movement. ReflecTouch [65] detected the grasping posture of smartphones by the reflection images of corneal. Regarding finger touch sensing, MirrorTrack [7] and SymmetriSense [60] mounted a near-surface camera to track the user's fingers together with its mirror reflection to detect multi-finger touch for specular surfaces.

Constructing fingers' shadow projection with customized light sources is another type of optical-enhanced strategy that has been explored by many research [24, 39, 40, 48, 50, 53, 54]. For example, Shoemaker et al. [47] and Song et al. [48] enhanced hand interaction on interactive display systems by recognizing hand shadows created from projector-screen casting. PlayAnywhere [54] presented an interactive projector system along with a shadow-based touch detection algorithm, allowing the user to perform touch input on the projected contents. ShadowSense [24] recognized social touch gestures between a human and a robot by positioning a camera behind the robot's translucent skin to capture shadows generated from human touch. Niikura et al. [39] constructed multi-source shadows by positioning multiple IR lights in the scene to indicate finger touch events with the shadow patterns captured by an infrared (IR) camera and two IR lights to detect the shadows of a finger. Another research of Niikura's [40] used wrist-worn LEDs and a camera to recognize finger-surface interaction and the fingers' moving directions. Although the wrist-emitted perspective was capable for observing the subtle contact state near fingers, such a system design required a camera worn between the wrist and the surface, leading the hand and the forearm to hover in the air to cause much fatigue and computational issues.

In our work, ShadowTouch adopted a wrist-emitted shadow projection strategy similar to Niikura's work [40]. Compared with prior work based on natural shadow projection (e.g., by a standard projector [47, 48, 54]), ShadowTouch took advantage of its unique optical design to achieve a larger amplification ratio of near-surface finger movements, thus having an increased spatial resolution in detecting hand-to-surface touches. Moreover, our design of constructing finger shadows with a wrist-mounted light source and reusing the cameras for hand tracking to enhance finger touch recognition allowed the user to interact with arbitrary surfaces with a light-weighted and computational-friendly hardware form.

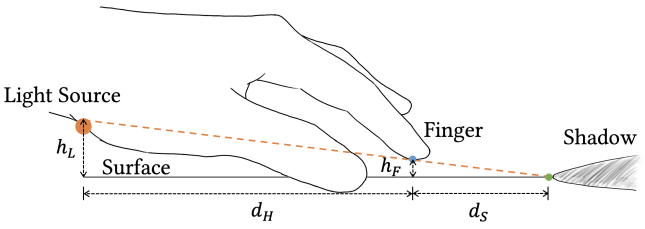

Figure 2: The working principle of ShadowTouch.

3 SHADOWTOUCH DESIGN

In this section, we first introduce the initial inspiration and the working principle of ShadowTouch. Then we explain the justifications for the optical design details to construct high-quality shadow features. Finally, we present the finalized hardware prototype.

3.1 Working Principle

ShadowTouch aims to construct recognizable optical features to indicate the accurate touch state of near-surface fingers, which is complementary to vision-based hand tracking in sensing subtle hand-to-surface gestures. The fundamental working principle is illustrated in Figure 2. A forward-facing light source L (e.g., a LED) is mounted on the user's wrist and lights up the neighboring area of the hand. When the hand is approaching a surface, some of the light cast on the surface is obstructed by fingers, thus yielding shadows in front of corresponding fingers. Assuming the height of the light source is h_L, and the height of the fingertip is h_F, the distance between the tip of the yielded shadow and the touch point d_S can be computed as:

$$d_S = \frac{d_H h_F}{(h_L - h_F)}$$
$$r_{mag} = \frac{d_S}{h_F} = \frac{d_H}{(h_L - h_F)} \quad (1)$$

, where d_H represents the projected distance between the light source and the fingertip on the surface, which is associated with the hand size and posture and is usually within a certain range (e.g., 10cm to 20cm). r_{mag} indicates the magnification ratio to measure how much ShadowTouch magnifies the subtle near-surface finger state by casting h_F to d_S. From the above formula, we noticed smaller h_L leads to a more significant magnification effect. Meanwhile, due to the reflection property of the diffuse surface, smaller h_L also means a smaller incident angle that causes less contrast of the shadow. Therefore, further justifications on the hardware form are needed to find an optimal solution balancing magnification and contrast to yield high-quality shadow features for recognition.

3.2 Optical Design Considerations

To facilitate the optical design in constructing high-quality shadow features, we discussed the design details by giving four research questions along with our explorations and considerations, as detailed below. Some of the considerations were generated from pilot experiments with three lab members.

DQ1: What are the general design goals for ShadowTouch hardware? Since the quality of the shadows largely depends on

the optical design, the general goal should be to yield high-quality shadow features. High-quality shadow features are expected to be: 1) With sufficient movement magnification ratio r_{mag}. We found pilot users hard to preserve the hover state when the finger was as close as $h_F < 1mm$ to the surface, so we assumed $1mm$ as the target resolution that ShadowTouch aimed to recognize, where the amplified shadow movement for the $1mm$ resolution should be salient for the camera view. 2) With recognizable contrast (or SNR). Since smaller h_L causes less contrast of the shadow, a minimum height of the light source should be determined to ensure the SNR of the shadow against environmental interference (e.g., the ambient light). 3) Covering rich gesture spaces. Shadows of different fingers should be separated and legible instead of being obstructed (e.g., the thumb would be obstructed by the thenar if the light was placed improperly) or fused with other hand segments. Also, although a general open-palm arced-finger posture is expected for ShadowTouch, different hand positions (e.g., wrist-resting and wrist hovering) should be considered and accommodated.

DQ2: Why the light source is mounted on the wrist instead of the fingers? Considering the form factor, the light source is most possibly to be deployed on the wrist combined with a smartwatch (or a wristband) or on the finger with a ring because smartwatches and rings are the most prevalent forms of wearable devices. We chose to mount the light source on the wrist instead of the fingers for two reasons: 1) The user usually performs a touch with his hand arced, leading to greater h_L and smaller d_H (see Section 3.1 for the definitions) - which indicates less magnification - for the finger-mounted setting. 2) The fingers are constantly moving or shaking, both intentionally (e.g., the touching finger) and unintentionally (e.g., the idle in-air fingers), during a touch, making the shadows volatile and inferior for recognition if the light source were mounted on a finger.

DQ3: How to determine the hyperparameters, including the height, the position, and the specification of the light source? The general guideline to optimize the form and the parameters of the light source is to optimize the three design goals in DQ 1. For shadow magnification, we found $d_S > 5mm$ showed good recognizability for a down-facing camera. Assuming $d_H = 120mm$ (e.g., for the index finger) averagely, we have $h_L < 25mm$ from Equation 1, which the height of the light source stand should be smaller than. For the shadow contrast, we found the height of the light should be at least $10mm$ so that the shadows can be seen by human eyes on an ordinary lightwood diffuse reflective desktop using a 0.06W LED light. Finally, considering the coverage of the gesture space, we found if the light source were squarely mounted on a wristband, the light would be easily blocked by the root of the palm, especially when the wrist rested on the surface. Therefore, we chose a design where the light source reaches out from the wrist to the root of the palm.

DQ4: What is the proper camera configuration used for ShadowTouch?

For simplicity, we did not consider a multi-camera solution for ShadowTouch. We adopted a head-mounted setting, where the camera was mounted on the top of the VR/AR headset with a down-facing perspective, as shown in Figure 3, mainly for two reasons. 1) Light-weighted hardware. Wrist-mounted cameras would introduce extra modules and circuits that not only increase the wristband's weight and size but also bring problems in balancing energy consumption, computation, and data transmission. As comparison, ShadowTouch merely requires a light widget instead of introducing an independent computing device. 2) Unified and reusable camera system. Although we used an external head-mounted camera in our prototype, the form factor is designed to be capable of reusing the original camera system of the HMD. The sensing pipeline can be seamlessly integrated into the original hand tracking system by reusing both the raw camera frames and the hand tracking results.

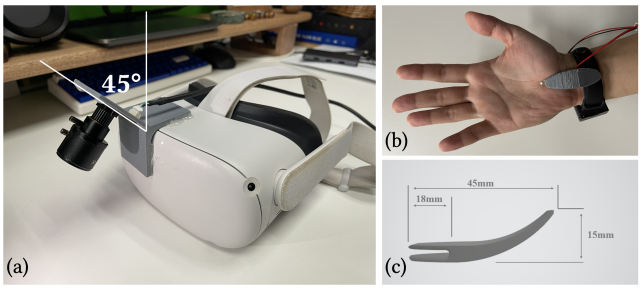

Figure 3: The hardware prototype of ShadowTouch. (a) The camera. (b) A wearing example of the wristband. (c) Specifications of the light widget.

To ensure the camera can capture the on-surface hand in a wide range, we chose a wide-range camera with 120° FoV. Regarding the frame rate, previous work [14] has shown a normal touch event is a transient process and typically happens within 50 ms. Using a 30-FPS camera in such a case would merely capture 1-2 frames and suffer severe motion blur for the whole touch process. Therefore, we chose a 120 FPS camera that can capture the whole process of touch, along with the movement of the shadows.

3.3 Hardware Prototype

We prototyped ShadowTouch hardware with a wearable wristband and a wide-range high-speed camera, as shown in Figure 3. We used a 0.06W LED bulb as the light source and fixed it on a 3D-printed plastic widget. The arc-shaped widget can be clamped on the wristband and reach out from the wrist to keep the light source at the root of the palm and 15mm above the desktop. The power supply of the light source is a 3V battery on the wristband. If needed, prototyping the widget with stretchable material (e.g., memory metal) can achieve better adaptability to different sizes of palms, hands, and various wearing habits.

The USB camera frame rate is 120 FPS, with a diagonal FoV of 120° (or a 113° horizontal FoV and a 81° vertical FoV), and a resolution of 1280×720. The camera is mounted on an Oculus Quest 2 VR headset with a down facing angle of 45°, as shown in Figure 3 (a).

4 SHADOWTOUCH ALGORITHMS

In this section, we introduced the sensing algorithms of ShadowTouch to recognize the hand-to-surface touch state of independent fingers.

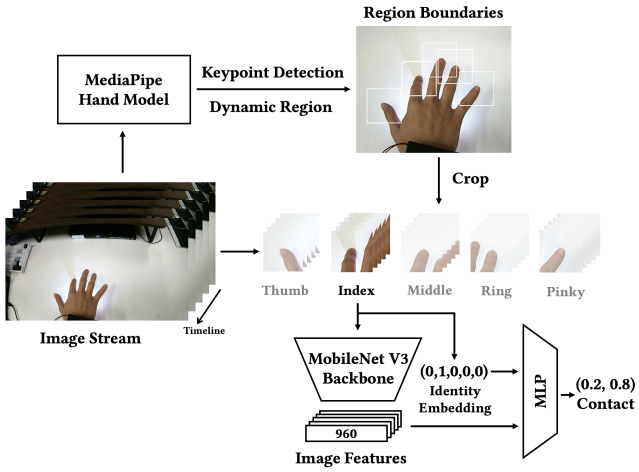

Figure 4: The algorithm pipeline of ShadowTouch.

4.1 Overall Pipeline

The overall algorithm pipeline is shown in Figure 4. Instead of directly training an end-to-end recognition model, we broke the task into two stages - fingertip region extraction and touch state recognition. The first stage detects the hand keypoints from an individual frame and outputs the boundaries of all fingertips with dynamic sizes. Then the second stage extracts cropped frame series within a short period and classifies the touch state for each finger. Considering the high-speed camera and the two stages were not necessarily to run at the same frequency (e.g., 120 FPS for the camera, but lower for the keypoint detection and touch detection), we designed an asynchronous workflow for our algorithms, where a group of shared boundaries of fingertip regions was maintained. Stage 1 would update the values of the boundaries at each computation, while stage 2 extracts the regions for recognition based on the latest boundaries.

4.2 Fingertip Region Extraction

We used the Mediapipe hand landmark model[63] to acquire the keypoints from the video frames. The model first detects palms from the image to find the valid hand regions and then performs a keypoint detection model to localize all 21 keypoints (which is a standard representation used in previous hand datasets [37]) for each hand. The position of each keypoint is a 3-D normalized coordinate where the z-axis represents the estimated depth relative to the root (wrist) joint. Based on the x- and y-coordinates output, we extracted the region boundaries of all five fingertips by using their keypoint coordinates as the centers and adopting a dynamic side length of $[\frac{D_X+D_Y}{a}] \cdot b$, where D_X and D_Y represents the size of the bounding box of all 21 hand keypoints in x- and y- axes respectively. a controls the discretization level and a, b control the region size. In our implementation, we used a camera with the resolution of 1280×720 and chose $a = 50$ and $b = 10$ based on a pilot real-time observation, ensuring an appropriate region size to capture complete shadow features while including the least other interfering areas.

4.3 Touch State Recognition from Fingertip Regions

To facilitate efficient touch state recognition, we built a light-weight CNN model based on a pre-trained MobileNet V3 backend [22], which was proven efficient in computation for mobile devices (e.g., running on the CPU of a smartphone). Given a series of consecutive k cropped frames $X_{i,0}, \cdots, X_{i,k-1}$ of finger i ($i = 0$ for the thumb and $i = 4$ for the pinky finger), our model first passed each $X_{i,s}$ through the MobileNet V3 backbone network to acquire a feature vector $\mathbf{v}_s$ with the shape of 960. Then we concatenated all the feature vectors, along with a 5-dimensional one-hot vector f_i to indicate the finger identity, and passed the concatenated vector $[\mathbf{v}_0, \cdots \mathbf{v}_{k-1}; f_i]$ through a multi-layer perceptron (MLP) classifier to acquire the touch state prediction. The whole model structure is shown in Figure 4. In real-time prediction, for a certain time step, we stacked all the cropped frames of 5 fingers into a batch to accelerate the forward computation of the MobileNet V3 backbone network.

For the training process, we adopted three data augmentation strategies - 1) randomly jitter the brightness and the hue by 0-0.5, 2) randomly rotate the image by $0 - 30^\circ$, and 3) randomly shift the image by 0-4 pixels - to improve the data diversity.

4.4 Implementation

The whole algorithm pipeline was implemented in Python and Pytorch on a Windows PC (CPU: Intel Core i9-12900KF; GPU Nvidia GeForce RTX 3090). The USB camera was wired to the PC, and the camera stream was read by a Python thread at 120 FPS. The keypoint detection and the touch recognition threads both ran at approximately 60 FPS. For a typical setting with a window size of 5, the pipeline reported the touch state of all the fingers at approximately 60 FPS with a delay of approximately 20 ms. The reported touch states were further streamed to a VR/AR device using web sockets to plug into the applications.

5 ALGORITHM EVALUATION

We conducted a systematic evaluation on our sensing pipeline and algorithms to gain an understanding of the performance in different settings and with different hyperparameters.

5.1 Participants and Apparatus

We recruited 12 participants (3 females), with an average age of 22.8 (SD=1.3), from the local campus by word-of-mouth. All participants were right-handed, and the sizes of their palms (from the root of the palm to the root of the middle finger) and hands (from the root of the palm to the tip of the middle finger) were 10.5 (SD=0.9) cm and 19.0 (SD=1.8) cm, respectively.

The settings of the wrist-mounted light source were the same as described in Sec 3.3 and Figure 3. We expected the collected video data could be automatically labeled with the minimum artifact, which implied two requirements for the data collection process: 1) the whole hand and the near-hand shadow region should always be fully captured in the camera view, and 2) the ground-truth touch signals should be simultaneously captured. For 1, to reduce the burden on participants, we mounted the camera on a fixed long arm gooseneck bracket for data collection, where the camera captured

Figure 5: Data collection settings. (a) Apparatus and setup. (b) An example of camera capture (cropped to highlight the hand region). (c) The selected background textures other than the white background.

the data from a perspective similar to the participant's eyes. The camera was randomly repositioned before the collection of each video segment to cover different camera perspectives. To achieve 2, a Sensel Morph multi-touch board was placed on the table at the bottom layer to collect ground-truth touch signals of different fingers used for automatic labeling. Moreover, we printed maps of three typical materials - white, lightwood, and marble (see Figure 5 (c)) - on paper and covered them on the Sensel board to simulate different kinds of planes to interact with. Figure 5 (a) showed an overview of the data collection environment, and Figure 5 (b) showed an example of camera capture (cropped to highlight the hand region).

5.2 Data Collection

The data collection process started with a brief introduction to the basic sensing goals and principles of ShadowTouch. Then the experimenter helped the participants put on the wristband and set up the camera and the Sensel board properly. Meanwhile, each participant's age and the size of their palms and hands were recorded.

The collection process contained 3 rounds. At the start of each round, a random background texture was picked and covered on the top of the Sensel board. Each participant was required to record 9 video segments in each round. They first performed 5 long taps (2-3 seconds for each tap) and 10 fast taps (≈0.5 second for each tap) with five individual fingers, respectively, using their right hand. Since the thumb, index finger, and middle finger are more commonly used in hand-to-surface gestures, they were required to perform 3 long taps, 3 double taps, and 3×4 slidings in four directions with these three fingers, respectively, to cover broader gesture spaces. Finally, they were instructed to collect a video of negative samples by either moving their fingers freely in the air, actively pretending to touch, or constructing other hard negative samples. Before the recording of each video, the camera was randomly repositioned by the experimenter to cover different camera perspectives. Participants repeated the above procedure to accomplish three rounds of data collection. A 30 seconds break was taken between two rounds. Each participant completed the study in around 20 minutes and was compensated for 10$.

As the result, we collected a total of 12 users $\times$ 3 background textures (white, lightwood, and marble) $\times$ 9 = 324 video segments. Each video segment lasted for approximately 25s (3000 video frames), and all the video data added up to approximately 135 minutes. In our data collection process, we have special considerations to cover different conditions, including camera position/perspective (by repositioning the camera in each video, equivalent to covering different regions), wrist height (not restricted), touching postures, textural background (three backgrounds), and ambient light (not restricted, collected in 3 rooms) to get more diverse and hard data.

During the recording, the camera captured the touch process along with the shadows from a perspective similar to the participant's eyes, while the Sensel board recorded the key frames on which the number of contact points changed. Since the identity of the touching finger for a certain video segment was unique and known, we ran an automatic labeling program to label the touch state (contact v.s. non-contact) of each finger for all video frames based on the data from the Sensel board.

5.3 Dataset and Evaluation Metrics

We constituted the datasets for training and evaluation by sampling frames from labeled video clips. We sampled N positive (contact) samples for the target finger and $5 \times N$ negative samples for all five fingers from each video with a positive label (e.g., index finger touch) while merely sampling $5 \times N$ negative samples for each finger from the negative videos. For the train set, we sparsely sampled video clips at a ratio of 5/1000, considering the memory load, and generated three datasets with different time window lengths of 1, 3, and 5 frames for each user, each containing approximately 400 positive samples and 2000 negative samples. For the test set, we sampled the video clips with the same strategy but a higher sampling ratio, generating a 5-frame datatset with approximately 1000 positive samples and 5000 negative samples for each user. To ensure the cross-user and cross-method consistency in the evaluation, we fixed the test datasets during the whole evaluation process, and for shorter time window lengths, test sets were generated by selecting the middle frames from the 5-frame samples.

Considering merely a small portion of frames were used for training in a sampling round, we developed an asynchronous data reloading strategy to generate more diverse training data and optimize data utilization efficiency. Specifically, we implemented the strategy by creating an individual thread that continuously updated the train sets with the same sparse sampling probability, while the main thread updated the data and conducted model training. The updated datasets were reloaded for every four training epochs.

We conducted leave-one-out cross-evaluations using two training strategies (cross-user, C-U; cross-user + asynchronously reloading train data, C-U-R) with three different window sizes (1, 3, 5 frames), and reported the accuracies, recalls, precisions, and F-1 scores of the touch state recognition for individual fingers and merged fingers.

5.4 Results

The means and standard deviations of recognition accuracy, precision, recall and F-1 score of touch state recognition for each finger across all participants under the leave-one-user-out setting was reported in Table 1. From the results, ShadowTouch generally achieved a promising performance in recognizing finger touch state, with an optimal average accuracy of 99.1% and an F-1 score

Table 1: Accuracies, precisions, recalls, and F-1 scores of our evaluation in two training strategies and three window sizes. Numbers in the table are in percentage (%).

Strategy	Finger	Window Size = 1				Window Size = 3				Window Size = 5			
		F-1	Acc	Rec	Prec	F-1	Acc	Rec	Prec	F-1	Acc	Rec	Prec
C-U	Thumb	97.2(2.3)	99.0(0.7)	97.7(3.3)	96.8(2.7)	96.7(2.0)	98.9(0.6)	97.2(2.3)	96.3(3.5)	96.7(2.9)	98.8(1.0)	97.8(3.3)	95.8(4.0)
	Index	96.3(3.5)	98.7(1.2)	95.8(5.6)	97.0(2.7)	97.3(2.9)	99.0(1.1)	97.8(2.8)	96.7(3.3)	96.5(2.9)	98.7(1.0)	98.3(1.4)	94.9(5.8)
	Middle	95.4(4.1)	98.4(1.3)	96.0(4.9)	95.0(5.0)	95.9(2.8)	98.6(0.9)	95.9(3.2)	95.9(3.4)	95.6(3.8)	98.4(1.4)	96.7(2.4)	94.9(7.1)
	Ring	92.3(7.0)	98.6(1.1)	92.0(11.4)	93.8(6.1)	92.4(6.5)	98.5(1.7)	91.3(7.5)	94.0(7.3)	90.4(9.7)	98.0(2.2)	93.9(8.5)	88.8(14.2)
	Pinky	88.1(10.7)	97.9(1.8)	86.8(12.9)	91.4(12.5)	88.5(13.0)	98.1(1.7)	86.6(17.4)	92.1(6.5)	86.3(15.6)	97.8(2.1)	86.0(18.8)	87.9(13.9)
	All	94.9(3.5)	98.5(0.9)	94.8(4.2)	95.2(4.0)	95.3(3.4)	98.6(1.0)	95.1(3.5)	95.5(3.7)	94.5(4.6)	98.4(1.3)	95.8(2.7)	93.5(7.2)
C-U-R	Thumb	97.4(2.3)	99.1(0.7)	98.0(1.9)	96.8(2.7)	97.2(2.1)	99.0(0.6)	97.0(3.1)	97.4(2.6)	98.1(1.7)	99.3(0.6)	98.9(2.1)	97.4(2.2)
	Index	96.6(2.8)	98.8(1.0)	97.5(3.0)	95.8(3.7)	96.6(2.7)	98.8(0.9)	96.7(3.5)	96.5(2.5)	98.0(2.1)	99.3(0.8)	98.6(2.2)	97.5(2.9)
	Middle	96.0(3.3)	98.6(1.1)	96.5(3.1)	95.6(4.7)	96.5(3.4)	98.7(1.2)	97.7(1.7)	95.4(5.5)	96.9(2.6)	98.9(0.8)	98.0(2.6)	96.0(4.2)
	Ring	91.0(7.7)	98.3(1.5)	90.2(11.6)	92.8(6.6)	90.8(9.2)	98.0(2.2)	92.6(7.5)	90.3(13.2)	94.0(5.1)	98.8(1.0)	93.6(7.6)	94.6(3.6)
	Pinky	89.3(10.1)	98.2(1.5)	87.2(15.1)	92.7(5.1)	89.2(10.7)	98.2(1.6)	87.3(13.1)	91.6(8.4)	94.0(5.6)	99.0(0.7)	93.7(7.3)	94.4(4.3)
	All	95.2(3.6)	98.6(0.9)	95.2(4.4)	95.3(3.3)	95.1(3.8)	98.6(1.0)	95.4(3.4)	94.8(4.8)	96.8(2.0)	99.1(0.6)	97.3(2.2)	96.4(2.3)

of 96.8% across all fingers using a 5-frame window and with the frame reloading strategy.

Regarding different window sizes, the optimal F-1 scores for 1-frame and 3-frame models were 95.2% and 95.1% respectively, which were lower than the F-1 score of 5-frame model. Friedman test found significant effects on window size ($\chi^2(2) = 10.17, p < 0.05$). We also observed that 5-frame model achieved more robust and stable recognition (e.g., with fewer trembles and transient errors) in real applications, probably because models with larger window sizes could observe the motion of fingers in a certain period rather than the static state in an individual frame, serving the role of a smoothing filter.

Regarding data reloading, we found it had greater improvements for 5-frame model (e.g., 96.7% v.s. 94.0%) rather than for 1-frame and 3-frame models, which was understandable because input data for larger window size should cover a larger sampling space containing temporal (e.g., motion-related) information. Wilcoxon Signed-Rank tests found significant effects on data reloading strategy for 5-frame model ($Z = -2.98, p < 0.05$), but no significant effect for 1-frame model ($Z = -0.55, p = 0.58$) and 3-frame model ($Z = -0.47, p = 0.64$). We also found the improvement even much stronger for the ring finger (94.0% v.s. 90.4%) and the pinky finger (94.0% v.s. 86.3%), probably because the ring and pinky finger regions were more frequently obstructed, thus with lower quality and greater noise.

The accuracies of different fingers were reported in Table 1. The optimal F-1 scores of the five fingers were 98.1%, 98.0%, 96.9%, 94.0%, and 94.0% respectively. The recognition results of thumb, index finger, and middle finger significantly outperformed that of ring finger and pinky finger (97.7% v.s. 94.0%, $Z = -2.34, p < 0.05$)

By manually examining the error cases, we found that the fingertip area in some samples was obviously wrong, which was caused by the error of Mediapipe as shown in Figure 6(b). We manually examined 5 (fingers) * 100 error samples, finding 13.8% of the errors were caused by Mediapipe (9, 16, 10, 20, and 14 for thumb to pinky finger respectively). Such errors could be potentially eliminated with a better hand tracking system. In addition, in the process of data collection, we found that ring and pinky fingers were easily blocked by the finger on their left when touching down (Figure 6(a)), and the flexibility of these two fingers varied greatly between

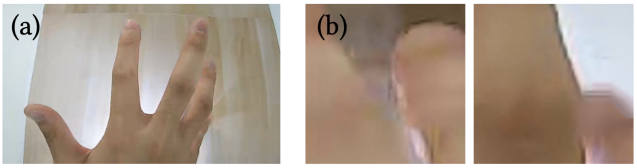

Figure 6: Examples of error cases: (a) The pinky finger is obstructed by other fingers. (b) Errors of MediaPipe fingertip recognition.

different participants. For some participants, it was difficult to avoid occlusion by controlling their fingers. In particular, the length of the little finger also varies greatly among the participants. Participants with longer little fingers tended to naturally bend their fingers and click, while those with shorter pinky fingers tended to straighten their fingers and click. As a result, the images cropped around fingertips vary, which also leads to a decline in recognition accuracy.

Here we also established an empirical comparison between ShadowTouch and state-of-the-art finger touch recognition techniques. As is most similar to ShadowTouch, PressureVision [12] presented a model to regress the hand's contact map with the surface in a one-frame bare-hand setting, achieving a contact accuracy of 90.4% for individual fingers. As a comparison, ShadowTouch achieved a finger contact accuracy of 98.6% for 1-frame model and 99.1% for 5-frame model, showing a significant improvement in accuracy and applicability, as well as the effectiveness of actively constructing shadow features to amplify near-surface finger state.

6 APPLICATIONS AND USABILITY EVALUATION

In this section, we discuss how ShadowTouch's sensing capability can benefit hand-to-surface interaction design and empower a large gesture space of free-form hand-to-surface interaction in mixed reality. We developed four application prototypes to showcase ShadowTouch's potential and applicability in real usage scenarios. Finally, we conducted a user study to evaluate the capability, user

experience, and subjective preferences between ShadowTouch and traditional hand-to-surface interaction schemes.

6.1 Design Space

ShadowTouch brings the unique capability of detecting the near-surface touch states of independent fingers, with which a broad hand-to-surface interaction space can be designed. Below we outlined ShadowTouch's interaction space from three aspects:

Touch-based interaction. Touch-based interaction is the most intuitive hand-to-surface interaction form that a user could take advantage of from the aligned haptic feedback. By deriving touch events from the touch states, ShadowTouch allows the user to perform touch on unmodified surfaces with different fingers, as previous work [14, 15, 34, 35, 64] supported. Moreover, a huge leap over existing solutions such as TapID [34, 35] is that since the touch states of different fingers were recognized independently, ShadowTouch inheritedly supports multi-touch interaction, which can be used for multi-touch shortcuts or touching multiple widgets simultaneously (e.g., in playing games).

Stroke-based interaction. Stroke-based interaction on the surface has a similar experience to performing finger swiping on a trackpad, which can be achieved by ShadowTouch from the capability of sensing the touch state of a certain finger. Combined with touch event detection, ShadowTouch can easily turn arbitrary surfaces into a touchpad interface that supports cursor control and keystroke events. Bringing multi-finger touch states to construct a multi-stroke interface could further improve gesture expressivity, where interactions based on multi-strokes, such as transforming an image, can be implemented. Although Shi et al. achieved sensing the touch state of a certain finger with IMUs [46], the gesture space was restricted because the system only recognized the touch state of one finger.

Out-of-surface information for interaction. Compared with the interaction space enabled by a touchscreen or touchpad, ShadowTouch has the unique benefit of combining the hand tracking results to provide out-of-surface information. For example, ShadowTouch indicates the identity of the touching finger so that different functions can be assigned to different fingers. Moreover, other information, such as the touching angle and the touching posture, can potentially be integrated into the interface (e.g., controlling the stroke size with the touching angle) to provide more expressive and intelligent interaction schemes.

6.2 Usage Scenarios and Application Prototypes

We developed four application prototypes - a home navigator, a photo album, a whiteboard, and a text editor to showcase the above-mentioned interaction concepts and functions. All four applications were developed using Unreal Engine (UE) 4.27 on an Oculus Quest 2 with the latest Hand Tracking 2.0[1] engine. At the start of usage, the user should put their index finger of the right hand on a physical surface and pinch with the thumb and the index finger to align the VR desktop with the real-world surface. Figure 7 showed an overview of the application prototypes, and the detailed functions of each application are described below.

Home Navigator. As shown in Figure 7, a home navigator was displayed on a virtual screen in front of the user. Three icons, corresponding to the following three applications, were shown on the navigator. The user can control a cursor as if using a virtual trackpad on the aligned surface. They can 1) swipe on the surface to move the cursor **(one-finger, stroke-based)** and 2) touch on the surface for confirmation **(one-finger, touch-based)**. When the user were in a certain application, they could perform a left- or right-swipe with the index and middle fingers to switch the application, or an upward swipe with the index and middle fingers to return to the home **(multi-finger, stroke-based)**. If the user were in the home, they could perform a downward swipe with the index and middle fingers to return to the last application.

Photo Album. As shown in Figure 7 (b), the photo album, displayed on the same virtual screen, allowed the user to swipe with the index finger to switch to different photos **(one-finger, stroke-based)**. For each photo, the user could touch and swipe using the thumb and the index finger to apply transformations (panning, zooming, and rotating) on the photo **(multi-finger, stroke-based)**.

Whiteboard. The whiteboard application shown in Figure 7 (c) consists of two UI regions - the main canvas at the center and the brush panel on the right. The user could assign different brushes to different fingers by tapping the corresponding brush size and color buttons on the brush panel with the target finger **(single-finger, touch-based, out-of-surface information)**. A brush cube indicating brush size and color was displayed above each finger. After assigning brushes to different fingers, the user could paint on the canvas with different fingers as if drawing with different brushes **(single-finger, stroke-based, out-of-surface information)**.

Text Editor. The text editor was shown in Figure 7 (d), consisting of a keyboard UI and a text input area. The user could tap on different keys with the index finger to perform key input **(single-finger, touch-based)**. They could also swipe on the keyboard with the middle finger **(single-finger, stroke-based, out-of-surface information)** to control the cursor's position.

6.3 Usability Evaluation

We conducted a brief usability evaluation study to validate ShadowTouch's advantages in real application scenarios.

6.3.1 Design. We compared ShadowTouch with two collision-based techniques enabled by Quest 2's built-in Hand Tracking 2.0 engine regarding the sensing resolution for near-surface fingertip movement and users' subjective experience in VR applications. The two baseline techniques applied 1) physically aligned interfaces (e.g., aligned to the desk, as described in Sec 6.2) and 2) hovering interfaces, respectively. Below we present the implementation details.

Collision-based touch detection: We acquire the positions of the predefined fingertip markers from Quest 2's hand tracking API [2] and set a vertical collision bar of 0.5 cm below and 2.0 cm above the fingertip landmark for each finger. The two thresholds were determined to maximize the probability of distinguishing the touch-down and 1-cm hovering states of the index finger for 100 surface alignment attempts in a pilot experiment. A touch event is triggered whenever the corresponding finger bar collision state

[1] https://developer.oculus.com/blog/presence-platforms-hand-tracking-api-gets-an-upgrade

[2] https://developer.oculus.com/documentation/unreal/unreal-hand-tracking

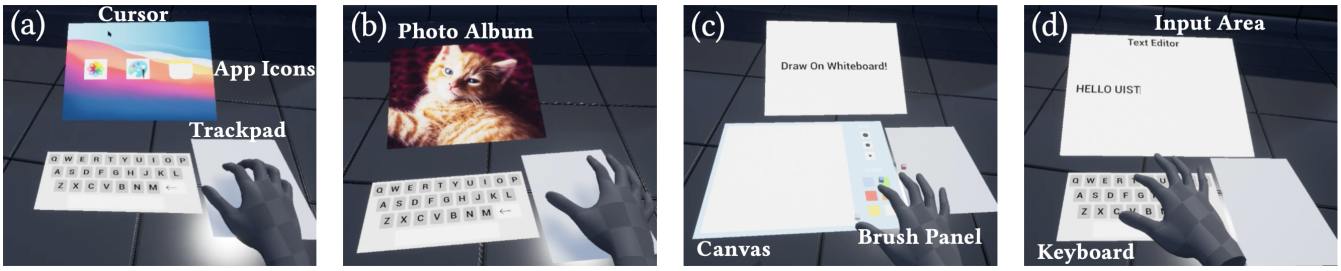

Figure 7: Application prototypes of ShadowTouch. (a) Home navigator. (b) Photo album. (c) Whiteboard. (d) Text Editor.

with the surface flips. Further optimization on avoiding unintended multi-finger touch is applied to each application, where multi-finger events are rejected if certain interaction is already in progress (e.g., if the cursor has been moved over a certain distance).

ShadowTouch: To further optimize the recognition stability of ShadowTouch, we further applied a state-based smoothing strategy to tackle the touch status fluctuation, based on our observation that realtime recognition pipeline occasionally yielded transient flippings (typically within 5 frames) on the touch state. When no finger touched down, 5 consecutive positive / negative frames were required to guarantee a touch-down / touch-up event. When a certain finger already touched down, we set a higher threshold of 10 consistent frames, which held a higher rejection rate to block both the recognition fluctuation and unintended touches to ensure the consistency of the currently performing gesture. The processed events were then sent to the VR application via socket.

6.3.2 Comparison in Sensing Resolution. To understand how subtle different techniques can distinguish the finger's vertical distance to the surface, we conducted a measurement to analyze the sensing resolution of each technique. For ShadowTouch, in a normal posture (camera perspective 45°, wrist ≈3cm hovering), the minimum height of the index fingertip from the surface with which ShadowTouch can stably report touch-up was around 2mm (measured by a millimeter scale at the same depth of the touch point from the recording of an along-surface macro camera). For collision-based algorithms, when the user touched the surface and slightly moved around their index finger in the same posture, the height variation of the fingertip reported by Quest 2 Hand Tracking 2.0 was 19.5mm (or 9.8mm for one lateral). Such a measurement validated that ShadowTouch could achieve a significantly higher resolution in differentiating subtle near-surface touch states compared with collision-based algorithms.

6.3.3 Subjective Experience in VR Applications. We recruited 10 participants (3 females, aged 23.7 (SD=1.3), with a familiarity score of 5.1 (SD=2.0) out of 7 for VR hand interaction) from the local campus to experience four VR Applications in three techniques described in Sec 6.3.1. The hardware settings and VR applications used in the study were the same as described in Sec 3.3 and Sec 6.2.

We designed interaction scripts to guide users' experience procedure. Specifically, we broke down each application into single-step interaction tasks according to the functions described in Sec 6.2 and designed scripts of a one-step atomic interaction sequence to form an integral usage flow. We summarized all possible atomic

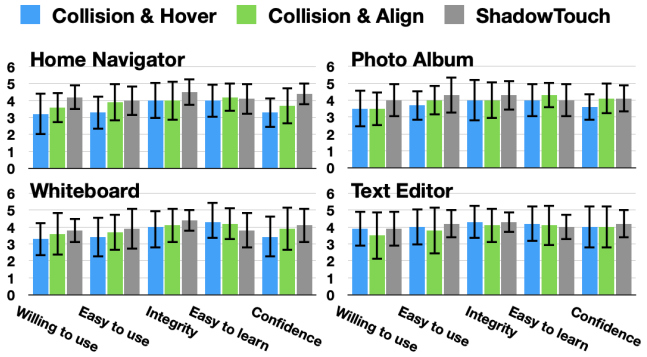

Figure 8: Subjective rating scores for different applications. 1 - strongly disagree, 5 - strongly agree.

interactions into three categories - one-finger touch, one-finger swipe, and multi-finger gesture - as shown in Table 2. The template scripts of different applications are shown in Table 3.

To evaluate users' experience and subjective ratings of different hand-to-surface interaction techniques, we designed a questionnaire containing five questions derived from the system usability scale (SUS) [4] on willingness to use, easiness to use, integrity, learnability, and confidence regarding different applications under three techniques. Users were further asked to provide their subjective feedback towards three different settings.

Figure 8 showed the subjective ratings on four applications under three techniques. Higher scores indicate more willing to use, easier to use, better integrity, more easy to learn, and higher confidence. The ratings indicated that participants generally favored the integration of ShadowTouch in all applications compared to other baseline techniques, which received a consistently higher score in willing to use (4.0), easy to use (4.1), integrity (4.3), and confidence (4.1). Most participants (N=8) mentioned aligning virtual interfaces to physical surfaces was an effective way of reducing physical burden. *"Aligned surface provided support of hand and better haptic feedback (P10)."* All participants showed a positive attitude towards ShadowTouch's potential leading new experiences in VR that had never been achieved before. *"Although the hand tracking seems quite satisfying, I can still feel the improvement brought by ShadowTouch in distinguishing subtle touch states (P1)." "The accuracy of the index and middle finger gestures was significantly higher than threshold methods. I was more confident in using the system*

(P8)." "ShadowTouch allowed me to touch with the rest fingers relaxed. I would not do this for threshold methods because it definitely causes many unintended errors (P2)." Meanwhile, some participants also saw concerns regarding the current version of implementations. *"When I dramatically moved my head, the tracking was cut off. I guess it was somewhat related to insufficient camera perspective (P5)." "The system should remind me when my hand moved out of the tracking area. Otherwise, it was a frustrating interruption (P10)."* For the easy-to-learn dimension, ShadowTouch received a lower score than threshold-based techniques, probably due to *"unclear trigger criterion"* and *"adaptation from the threshold-based logic"* from users' comments. *"It is hard for me to imagine how ShadowTouch detects the touches (P10)." "In my experience, hand gestures should be exaggerated in VR for better performance. It is a conceptual change (P3)."*

7 LIMITATIONS AND DISCUSSION

In this section, we discuss the limitations and potential considerations related to the practical deployment of ShadowTouch.

7.1 System Robustness

Although ShadowTouch generally showed good performance in our work, since it adopted a vision-based sensing solution, it inevitably yields failure cases due to the inherited drawbacks of the camera, such as improper ambient light (e.g., too bright or too dark for hand t racking), structural obstruction, peripheral camera view, etc. For example, in our evaluation, we found the ring finger and the pinky finger had lower recognition accuracy because the fingertips and the shadows were occasionally obstructed by the dorsum or other fingers. From the shadow side, the contrast (or SNR) of the shadow was largely confined by the material of the surface. For example, a reflective surface or a black surface (that absorbs most of the light) would lead to a significantly lower SNR from the camera's view.

To alleviate these problems and enhance the system robustness, possible solutions include: 1) compensating the ambient light with an active light source, 2) fusing camera data from different views (e.g., a global view or a third-person view) to reduce obstruction, and 3) adopting a controllable wrist light source to construct additional optical features (e.g., the flicker of the light could be encoded or synchronized with the camera frequency to acquire differential features between frames). All these potential solutions are worthy of further research to improve the system robustness of ShadowTouch.

Moreover, further research on adapting ShadowTouch on different shapes of surfaces (e.g., vertical surfaces or surfaces with irregular geometries) is worthwhile. From our empirical observation, touching vertical surfaces yielded similar shadow features to the samples with high-raised wrist (4cm), open palm, and large camera incident angle (e.g., >60 degree) in our dataset, which is with least obstruction and easy to recognize. Irregular geometries would lead to partial deformation and artifacts of the shadow features. We believe collecting more diverse data and conducting more detailed evaluations regarding different surfaces is of great practical value.

7.2 Form Factor and Power Consumption

Currently, ShadowTouch is implemented in a single-hand mode with a visible light source, mainly to validate the computational feasibility of the optical design and the algorithms. In our prototype, we simplify the setting with a normal LED, considering: 1) the convenience of debugging and monitoring with shadows and 2) no need for customization of the camera. In our implementation, we did not have a special design to alleviate the ambient light interference (e.g., the shadows from ambient light near fingertips existed for a portion of samples), mainly to improve the robustness of our model against ambient light interference. We believe adopting invisible light could further improve the system's performance and be more applicable for AR scenarios for not disturbing the user's attention. Also, further work on modifying ShadowTouch into a two-hand version is essential to achieve better usability with the support of two-handed interaction.

Regarding the computational cost, currently, we prototyped our algorithms on a PC with strong computational performance to guarantee the pipeline running at a constant frame rate. Our results have shown that light-weighted models chosen for our pipeline (total params for two stages less than 10M) could achieve a promising performance. Moreover, reusing the hand tracking results of the HMD to substitute the hand tracking stage in our pipeline could further reduce computation. Therefore, we believe ShadowTouch is capable of deployment on commodity VR and AR devices. Further optimizations such as quantization on the parameters [8, 9] could further reduce computational cost.

8 CONCLUSION

We present ShadowTouch, a novel vision-based sensing technique to recognize the touch states with a surface for individual fingers by constructing recognizable shadow features that magnify the subtle near-surface finger movement with a wrist-mounted light source. Taking advantage of ShadowTouch's unique optical design, the subtle vertical movements of near-surface fingers are cast to shadows on the surface, which are highly recognizable for the event a light-weight computer vision model. We also discussed and demonstrated the applicability of ShadowTouch by outlining the interaction space and developing application prototypes. The uniqueness of ShadowTouch sheds light on the following two aspects: 1) we presented a vision-based solution to recognize the near-surface touch state for individual fingers enhanced by wrist-projected shadows, reaching a cross-user accuracy of 99.1% and an F-1 score of 96.8%, which showed good applicability in our usability evaluation study; and 2) through the practice of ShadowTouch, we demonstrated such a methodology of auxiliary optical feature construction is simple but effective in exploiting the potentials of vision-based sensing techniques. We believe our work would play an essential role in facilitating physical-aligned hand interactions for mixed reality in the future.

ACKNOWLEDGMENTS

This work is supported by the Natural Science Foundation of China under Grant No. 62132010, and by Beijing Key Lab of Networked Multimedia, the Institute for Guo Qiang, Tsinghua University, Institute for Artifcial Intelligence, Tsinghua University (THUAI), and by 2025 Key Technological Innovation Program of Ningbo City under Grant No. 2022Z080, and by Beijing Municipal Science & Technology Commission, Administrative Commission of Zhongguancun Science Park No.Z221100006722018.

REFERENCES

[1] 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)*. 197–200. https://doi.org/10.1109/TABLETOP.2007.29

[2] Rahul Arora, Rubaiat Habib Kazi, Tovi Grossman, George Fitzmaurice, and Karan Singh. 2018. SymbiosisSketch: Combining 2D & 3D Sketching for Designing Detailed 3D Objects in Situ. 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, 1–15. https://doi.org/10.1145/3173574.3173759

[3] Rahul Arora, Rubaiat Habib Kazi, Fraser Anderson, Tovi Grossman, Karan Singh, and George Fitzmaurice. 2017. Experimental Evaluation of Sketching on Surfaces in VR. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 5643–5654. https://doi.org/10.1145/3025453.3025474

[4] Aaron Bangor, Philip T Kortum, and James T Miller. 2008. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction* 24, 6 (2008), 574–594.

[5] Samarth Brahmbhatt, Cusuh Ham, Charles C. Kemp, and James Hays. 2019. ContactDB: Analyzing and Predicting Grasp Contact via Thermal Imaging. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 8701–8711. https://doi.org/10.1109/CVPR.2019.00891

[6] Liwei Chan, Rong-Hao Liang, Ming-Chang Tsai, Kai-Yin Cheng, Chao-Huai Su, Mike Y. Chen, Wen-Huang Cheng, and Bing-Yu Chen. 2013. FingerPad: Private and Subtle Interaction Using Fingertips. 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, 255–260. https://doi.org/10.1145/2501988.2502016

[7] Pak-Kiu Chung, Bing Fang, and Francis Quek. 2008. Mirrortrack-a vision based multi-touch system for glossy display surfaces. (2008).

[8] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. 2014. Training deep neural networks with low precision multiplications. arXiv:1412.7024 [cs.LG]

[9] Tim Dettmers. 2015. 8-Bit Approximations for Parallelism in Deep Learning. arXiv:1511.04561 [cs.NE]

[10] Tobias Drey, Jan Gugenheimer, Julian Karlbauer, Maximilian Milo, and Enrico Rukzio. 2020. VRSketchIn: Exploring the Design Space of Pen and Tablet Interaction for 3D Sketching in Virtual 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/10.1145/3313831.3376628

[11] Cathy Mengying Fang and Chris Harrison. 2021. Retargeted Self-Haptics for Increased Immersion in VR without Instrumentation. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 1109–1121. https://doi.org/10.1145/3472749.3474810

[12] Patrick Grady, Chengcheng Tang, Samarth Brahmbhatt, Christopher D Twigg, Chengde Wan, James Hays, and Charles C Kemp. 2022. PressureVision: Estimating Hand Pressure from a Single RGB Image. *arXiv preprint arXiv:2203.10385* (2022).

[13] Patrick Grady, Chengcheng Tang, Samarth Brahmbhatt, Christopher D. Twigg, Chengde Wan, James Hays, and Charles C. Kemp. 2022. PressureVision: Estimating Hand Pressure from a Single RGB Image. In *Computer Vision – ECCV 2022*, Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (Eds.). Springer Nature Switzerland, Cham, 328–345.

[14] 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. https://doi.org/10.1145/3332165.3347947

[15] Yizheng Gu, Chun Yu, Zhipeng Li, Zhaoheng Li, Xiaoying Wei, and Yuanchun Shi. 2020. QwertyRing: Text Entry on Physical Surfaces Using a Ring. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 4, Article 128 (dec 2020), 29 pages. https://doi.org/10.1145/3432204

[16] Hung-Jui Guo and Balakrishnan Prabhakaran. 2022. HoloLens 2 Technical Evaluation as Mixed Reality Guide. https://doi.org/10.48550/ARXIV.2207.09554

[17] Sean Gustafson, Christian Holz, and Patrick Baudisch. 2011. Imaginary Phone: Learning Imaginary Interfaces by Transferring Spatial Memory from a Familiar Device. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (Santa Barbara, California, USA) *(UIST '11)*. Association for Computing Machinery, New York, NY, USA, 283–292. https://doi.org/10.1145/2047196.2047233

[18] 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

[19] Chris Harrison, Hrvoje Benko, and Andrew D. Wilson. 2011. OmniTouch: Wearable Multitouch Interaction Everywhere. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (Santa Barbara, California, USA) *(UIST '11)*. Association for Computing Machinery, New York, NY, USA, 441–450. https://doi.org/10.1145/2047196.2047255

[20] Chris Harrison and Scott E. Hudson. 2008. Scratch Input: Creating Large, Inexpensive, Unpowered and Mobile Finger Input Surfaces. 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, 205–208. https://doi.org/10.1145/1449715.1449747

[21] Chris Harrison, Desney Tan, and Dan Morris. 2010. Skinput: Appropriating the Body as an Input Surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Atlanta, Georgia, USA) *(CHI '10)*. Association for Computing Machinery, New York, NY, USA, 453–462. https://doi.org/10.1145/1753326.1753394

[22] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. 2019. Searching for MobileNetV3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[23] Tzu Hsuan Hsia, Shogo Okamoto, Yasuhiro Akiyama, and Yoji Yamada. 2021. HumTouch: Localization of Touch on Semi-Conductive Surfaces by Sensing Human Body Antenna Signal. *Sensors* 21, 3 (2021). https://doi.org/10.3390/s21030859

[24] Yuhan Hu, Sara Maria Bejarano, and Guy Hoffman. 2020. ShadowSense: Detecting Human Touch in a Social Robot Using Shadow Image Classification. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 4, Article 132 (dec 2020), 24 pages. https://doi.org/10.1145/3432202

[25] Ying Jiang, Congyi Zhang, Hongbo Fu, Alberto Cannavò, Fabrizio Lamberti, Henry Y K Lau, and Wenping Wang. 2021. HandPainter - 3D Sketching in VR with Hand-Based Physical Proxy. 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 412, 13 pages. https://doi.org/10.1145/3411764.3445302

[26] Yongkwan Kim, Sang-Gyun An, Joon Hyub Lee, and Seok-Hyung Bae. 2018. Agile 3D Sketching with Air Scaffolding. 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, 1–12. https://doi.org/10.1145/3173574.3173812

[27] Yuki Kubo, Yuto Koguchi, Buntarou Shizuki, Shin Takahashi, and Otmar Hilliges. 2019. AudioTouch: Minimally Invasive Sensing of Micro-Gestures via Active Bio-Acoustic Sensing. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services* (Taipei, Taiwan) *(MobileHCI '19)*. Association for Computing Machinery, New York, NY, USA, Article 36, 13 pages. https://doi.org/10.1145/3338286.3340147

[28] Eric Larson, Gabe Cohn, Sidhant Gupta, Xiaofeng Ren, Beverly Harrison, Dieter Fox, and Shwetak Patel. 2011. HeatWave: Thermal Imaging for Surface User Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. Association for Computing Machinery, New York, NY, USA, 2565–2574. https://doi.org/10.1145/1978942.1979317

[29] Chen Liang, Chi Hsia, Chun Yu, Yukang Yan, Yuntao Wang, and Yuanchun Shi. 2023. DRG-Keyboard: Enabling Subtle Gesture Typing on the Fingertip with Dual IMU Rings. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 4, Article 170 (jan 2023), 30 pages. https://doi.org/10.1145/3569463

[30] Chen Liang, Chun Yu, Yue Qin, Yuntao Wang, and Yuanchun Shi. 2021. DualRing: Enabling Subtle and Expressive Hand Interaction with Dual IMU Rings. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 3, Article 115 (sep 2021), 27 pages. https://doi.org/10.1145/3478114

[31] Guan Ming Lim, Prayook Jatesiktat, Christopher Wee Keong Kuah, and Wei Tech Ang. 2020. Camera-based Hand Tracking using a Mirror-based Multi-view Setup. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. 5789–5793. https://doi.org/10.1109/EMBC44109.2020.9176728

[32] Damien Masson, Alix Goguey, Sylvain Malacria, and Géry Casiez. 2017. WhichFingers: Identifying Fingers on Touch Surfaces and Keyboards Using Vibration Sensors. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City, QC, Canada) *(UIST '17)*. Association for Computing Machinery, New York, NY, USA, 41–48. https://doi.org/10.1145/3126594.3126619

[33] 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

[34] Manuel Meier, Paul Streli, Andreas Fender, and Christian Holz. 2021. Demonstrating the Use of Rapid Touch Interaction in Virtual Reality for Prolonged Interaction in Productivity Scenarios. In *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. 761–762. https:

//doi.org/10.1109/VRW52623.2021.00263
[35] 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. https://doi.org/10.1109/VR50410.2021.00076
[36] Pranav Mistry and Pattie Maes. 2011. Mouseless: A Computer Mouse as Small as Invisible. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI EA '11)*. Association for Computing Machinery, New York, NY, USA, 1099–1104. https://doi.org/10.1145/1979742.1979715
[37] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. 2020. Interhand2. 6m: A dataset and baseline for 3d interacting hand pose estimation from a single rgb image. In *European Conference on Computer Vision*. Springer, 548–564.
[38] Anh Nguyen and Amy Banic. 2015. 3DTouch: A wearable 3D input device for 3D applications. In *2015 IEEE Virtual Reality (VR)*. 55–61. https://doi.org/10.1109/VR.2015.7223324
[39] Takehiro Niikura, Takashi Matsubara, and Naoki Mori. 2016. Touch Detection System for Various Surfaces Using Shadow of Finger. 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, 337–342. https://doi.org/10.1145/2992154.2996777
[40] 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. https://doi.org/10.1145/2582051.2582090
[41] Ju Young Oh, Jun Lee, Joong Ho Lee, and Ji Hyung Park. 2017. AnywhereTouch: Finger Tracking Method on Arbitrary Surface Using Nailed-Mounted IMU for Mobile HMD. In *HCI International 2017 – Posters' Extended Abstracts*, Constantine Stephanidis (Ed.). Springer International Publishing, Cham, 185–191.
[42] 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. https://doi.org/10.1109/ACCESS.2020.2997972
[43] Joseph A Paradiso and Che King Leo. 2005. Tracking and characterizing knocks atop large interactive displays. *Sensor Review* (2005).
[44] Luis Paredes, Ananya Ipsita, Juan C. Mesa, Ramses V. Martinez Garrido, and Karthik Ramani. 2022. StretchAR: Exploiting Touch and Stretch as a Method of Interaction for Smart Glasses Using Wearable Straps. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 3, Article 134 (sep 2022), 26 pages. https://doi.org/10.1145/3550305
[45] Yuto Sekiya, Takeshi Umezawa, and Noritaka Osawa. 2021. Detection of Finger Contact with Skin Based on Shadows and Texture Around Fingertips. In *Human-Computer Interaction. Interaction Techniques and Novel Applications: Thematic Area, HCI 2021, Held as Part of the 23rd HCI International Conference, HCII 2021, Virtual Event, July 24–29, 2021, Proceedings, Part II 23*. Springer, 109–122.
[46] Yilei Shi, Haimo Zhang, Kaixing Zhao, Jiashuo Cao, Mengmeng Sun, and Suranga Nanayakkara. 2020. Ready, Steady, Touch! Sensing Physical Contact with a Finger-Mounted IMU. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 2, Article 59 (June 2020), 25 pages. https://doi.org/10.1145/3397309
[47] Garth Shoemaker, Anthony Tang, and Kellogg S. Booth. 2007. Shadow Reaching: A New Perspective on Interaction for Large Displays. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology* (Newport, Rhode Island, USA) *(UIST '07)*. Association for Computing Machinery, New York, NY, USA, 53–56. https://doi.org/10.1145/1294211.1294221
[48] Chengqun Song and Jun Cheng. 2017. A robust projector–camera interactive display system based on finger touch control by fusing finger and its shadow. *Journal of the Society for Information Display* 25, 9 (2017), 568–576.
[49] Naoki Sugita, Daisuke Iwai, and Kosuke Sato. 2008. Touch sensing by image analysis of fingernail. In *2008 SICE Annual Conference*. 1520–1525. https://doi.org/10.1109/SICE.2008.4654901
[50] Joseph Thomas. 2013. A Camera Based Virtual Keyboard with Touch Detection by Shadow Analysis. (2013).
[51] Cheng-Yao Wang, Min-Chieh Hsiu, Po-Tsung Chiu, Chiao-Hui Chang, Liwei Chan, Bing-Yu Chen, and Mike Y Chen. 2015. Palmgesture: Using palms as gesture interfaces for eyes-free input. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 217–226.
[52] Eric Whitmire, Mohit Jain, Divye Jain, Greg Nelson, Ravi Karkar, Shwetak Patel, and Mayank Goel. 2017. DigiTouch: Reconfigurable Thumb-to-Finger Input and Text Entry on Head-Mounted Displays. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article 113 (sep 2017), 21 pages. https://doi.org/10.1145/3130978
[53] Andrew D Wilson. 2004. TouchLight: an imaging touch screen and display for gesture-based interaction. In *Proceedings of the 6th international conference on Multimodal interfaces*. 69–76.
[54] Andrew D Wilson. 2005. PlayAnywhere: a compact interactive tabletop projection-vision system. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*. 83–92.
[55] 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 10, 5 pages. https://doi.org/10.1145/2999508.2999522
[56] M. Yang, M. Al-Kutubi, and D.T. Pham. 2013. Continuous acoustic source tracking for tangible acoustic interfaces. *Measurement* 46, 3 (2013), 1272–1278. https://doi.org/10.1016/j.measurement.2012.11.019
[57] 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. https://doi.org/10.1145/2380116.2380137
[58] 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
[59] Xin Yi, Chen Liang, Haozhan Chen, Jiuxu Song, Chun Yu, Hewu Li, and Yuanchun Shi. 2023. From 2D to 3D: Facilitating Single-Finger Mid-Air Typing on QWERTY Keyboards with Probabilistic Touch Modeling. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 1, Article 38 (mar 2023), 25 pages. https://doi.org/10.1145/3580829
[60] Chungkuk Yoo, Inseok Hwang, Eric Rozner, Yu Gu, and Robert F. Dickerson. 2016. SymmetriSense: Enabling Near-Surface Interactivity on Glossy Surfaces Using a Single Commodity Smartphone. 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, 5126–5137. https://doi.org/10.1145/2858036.2858286
[61] Takatoshi Yoshida, Junichi Ogawa, Kyung Yun Choi, Sanad Bushnaq, Ken Nakagaki, and Hiroshi Ishii. 2021. InDepth: Force-Based Interaction with Objects beyond A Physical Barrier. In *Proceedings of the Fifteenth International Conference on Tangible, Embedded, and Embodied Interaction* (Salzburg, Austria) *(TEI '21)*. Association for Computing Machinery, New York, NY, USA, Article 42, 6 pages. https://doi.org/10.1145/3430524.3442447
[62] 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
[63] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. 2020. MediaPipe Hands: On-device Real-time Hand Tracking. https://doi.org/10.48550/ARXIV.2006.10214
[64] Mingrui Ray Zhang, Shumin Zhai, and Jacob O. Wobbrock. 2022. TypeAnywhere: A QWERTY-Based Text Entry Solution for Ubiquitous Computing. 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 339, 16 pages. https://doi.org/10.1145/3491102.3517686
[65] Xiang Zhang, Kaori Ikematsu, Kunihiro Kato, and Yuta Sugiura. 2022. ReflecTouch: Detecting Grasp Posture of Smartphone Using Corneal Reflection Images. In *CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 289, 8 pages. https://doi.org/10.1145/3491102.3517440
[66] Yang Zhang, Wolf Kienzle, Yanjun Ma, Shiu S. Ng, Hrvoje Benko, and Chris Harrison. 2019. ActiTouch: Robust Touch Detection for On-Skin AR/VR Interfaces. 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, 1151–1159. https://doi.org/10.1145/3332165.3347869

A APPENDIX

Table 2: Descriptions of atomic interactions in four applications and their categorization.

Index	Application	Atomic Interaction Description	Gesture Type
1	Home Navigator	move cursor with index finger	single-finger swipe
2	Home Navigator	click on an APP icon with index finger	single-finger touch
3	Home Navigator	swipe index finger and middle finger up to return to the home	multi-finger gesture
4	Home Navigator	swipe index finger and middle finger down to return to the last app	multi-finger gesture
5	Home Navigator	swipe index finger and middle finger left/right to switch app	multi-finger gesture
6	Photo Album	swipe index finger left/right to switch to next/previous images	single-finger swipe
7	Photo Album	move thumb and index finger to zoom/rotate/pan the image.	multi-finger gesture
8	Whiteboard	tap with different fingers to choose color and brush size	single-finger touch
9	Whiteboard	draw/erase with different fingers	single-finger swipe
10	Text Editor	Type with index finger	single-finger touch
11	Text Editor	move cursor with middle finger	single-finger swipe

Table 3: The template scripts used by the experimenter to guide the user study tasks in four applications.

Application	Instruction
Application 1: Home Navigator	1. Move the cursor with index finger over a [APP Name] app icon. 2. Tap with index finger on the [APP Name] app icon to launch the app. 3. Swipe up with index finger and middle finger to return to the home. 4. Swipe down with index finger and middle finger to return to the last opened app. 5. Swipe left/right with index finger and middle finger to switch to the next/previous app.
Application 2: Photo Album	1. Swipe left/right with index finger on trackpad to switch to the next/previous photo. 2. Move thumb and index finger on trackpad to zoom/rotate/pan the image. 3. Find the image of [Animal]. 4. Zoom in on [Body Part] of the [Animal].
Application 3: Whiteboard	1. Thumb, index finger, and middle finger can all be used for painting. Each finger can be bound with different brush style. Brush style will be displayed in the form of cubes floating around fingertips. 2. Tap on control panel to choose brush size and color for the touching finger. 3. Choose [Color] with index finger. 4. Draw [Object1] in the middle of the canvas. 5. Choose [Color] with middle finger. 6. Choose [Brush Size] with middle finger. 7. Draw [Object2] with middle finger. 8. Choose the eraser with thumb. 9. Erase [Object1] or [Object2].
Application 4: Text Editor	1. Type with index finger. 2. Move the cursor by swiping middle finger leftwards/rightwards on keyboard. 3. Enter "[Word1] [Word2]" in the input box. 4. Move the cursor between two words. 5. Delete "[Word1]". 6. Enter "[Word3]" at the beginning of the line.