

EclipseTouch: Touch Segmentation on Ad Hoc Surfaces using Worn Infrared Shadow Casting

Vimal Mollyn*

Carnegie Mellon University
Pittsburgh, PA, USA
vmollyn@cs.cmu.edu

Nathan DeVrio*

Carnegie Mellon University
Pittsburgh, PA, USA
ndevrio@cmu.edu

Chris Harrison

Carnegie Mellon University
Pittsburgh, PA, USA
chris.harrison@cs.cmu.edu

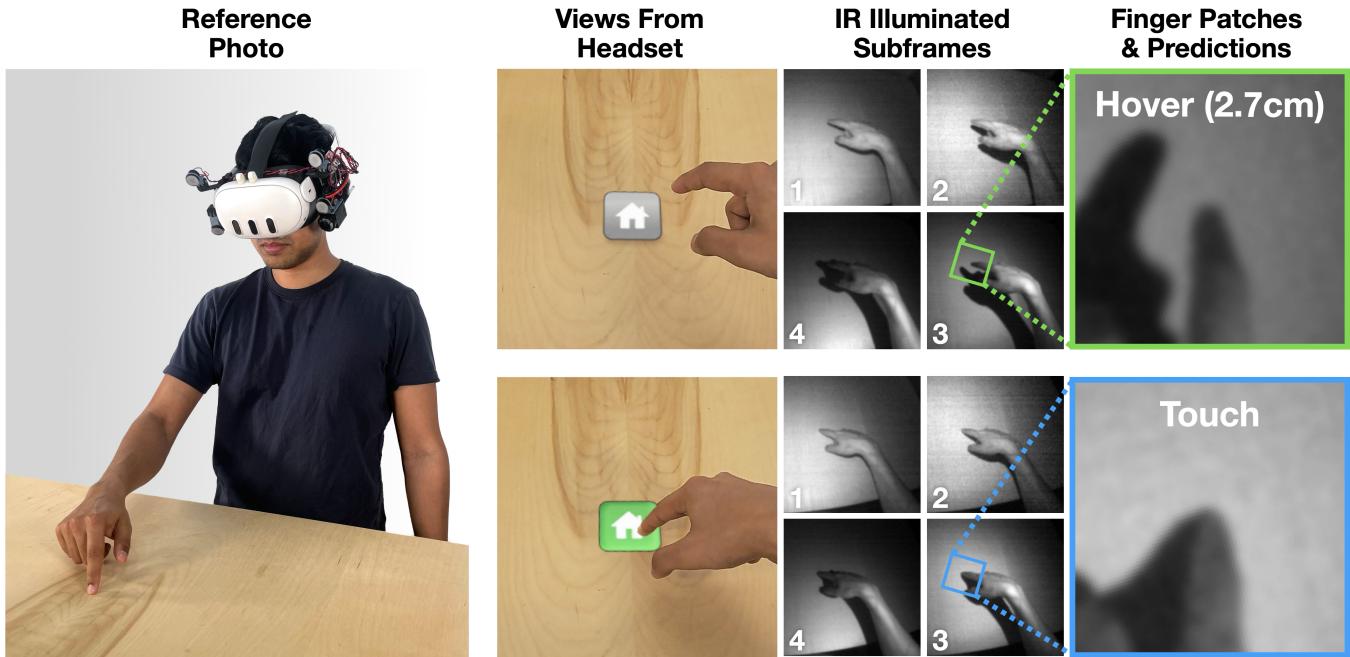


Figure 1: EclipseTouch is a headset-integrated sensing approach for touch input on ad hoc surfaces. The headset illuminators create structured shadows in infrared (1/2/3/4), which our system uses to estimate touch contact and hover distance.

Abstract

The ability to detect touch events on uninstrumented, everyday surfaces has been a long-standing goal for mixed reality systems. Prior work has shown that virtual interfaces bound to physical surfaces offer performance and ergonomic benefits over tapping at interfaces floating in the air. A wide variety of approaches have been previously developed, to which we contribute a new headset-integrated technique called EclipseTouch. We use a combination of a computer-triggered camera and one or more infrared emitters to create structured shadows, from which we can accurately estimate hover distance (mean error of 6.9 mm) and touch contact (98.0%

accuracy). We discuss how our technique works across a range of conditions, including surface material, interaction orientation, and environmental lighting.

CCS Concepts

- Human-centered computing → Mixed / augmented reality; Gestural input; Touch screens;
- Computing methodologies → Computer vision.

Keywords

Computer Vision, Input Techniques, Touch Surfaces and Touch Interaction, Virtual/Augmented Reality

*Both authors contributed equally.



This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

UIST '25, Busan, Republic of Korea

© 2025 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-2037-6/2025/09

<https://doi.org/10.1145/3746059.3747743>

ACM Reference Format:

Vimal Mollyn, Nathan DeVrio, and Chris Harrison. 2025. EclipseTouch: Touch Segmentation on Ad Hoc Surfaces using Worn Infrared Shadow Casting. In *The 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25), September 28–October 1, 2025, Busan, Republic of Korea*. ACM, New York, NY, USA, 14 pages. <https://doi.org/10.1145/3746059.3747743>

1 Introduction

Mixed reality (XR/AR) headsets are becoming more widespread, with increasing consumer excitement about forthcoming glasses-like form factors. However, navigating user interfaces on these devices generally requires users either to carry accessory controllers everywhere they go, or be limited to poking and swiping at interfaces in the air.

A complementary, long-envisioned interaction option is to ground virtual interfaces to real-world surfaces. One of the most reliable ways to achieve such interactions is to instrument surfaces with sensors. While robust, it is not feasible to do this for every surface in the world. In a future with highly mobile wearers of AR glasses, users may wish to temporarily and opportunistically appropriate surfaces for touch interaction. For this reason, there exists a significant body of work on ad hoc touch input without instrumentation of the environment. In most cases, this requires instrumenting the user instead, often with a special-purpose accessory device. Even when such sensors could be plausibly integrated into future smartwatches, it is most common for people to wear watches on their non-dominant hand.

We believe the ideal sensing method should integrate directly into the headset/glasses so that the user only needs to carry a single, self-contained device (i.e., the glasses). Only a handful of methods, which we discuss in Related Work, achieve this property. To this, we add two additional practical constraints: robust operation 1) across a wide variety of materials, including the user's skin; and 2) across environmental conditions (dark, bright, noisy, moving, etc.).

In this work, we present EclipseTouch, a new headset-integrated technique for ad hoc touch sensing. Our system leverages the well-known phenomena of shadow casting, utilized in prior touch sensing work, but never demonstrated in a single worn device. More specifically, EclipseTouch uses an infrared, egocentric headset camera, which captures shadows cast by one or more synchronized infrared illuminators on the headset (Figure 2). As the geometry between the camera and illuminators are fixed, these shadows inherently capture a finger's distance from a surface, including direct contact (Figure 11). Importantly, our approach must first filter out shadows cast by extraneous light sources in order to be robust. At the core of our system is an optimized deep neural network that has an inference time of 0.47 ms on an Apple M2 processor (used in the Apple Vision Pro). The result is a method that works "out of the box" and requires no pre-registration or calibration of the environment, surface, or user. Our approach works across a wide array of common surfaces, as well as lighting conditions, from bright to pitch-black. EclipseTouch can also be readily integrated into several popular XR headsets that already contain the requisite sensing hardware and compute. Taken together, this set of capabilities sets it apart from prior work, even those relying on similar phenomena.

2 Related Work

In this section, we review prior systems that have used techniques relevant to EclipseTouch. We begin with systems that examined the problem of ad hoc surface touch detection. For these, we start with instrumented environments (the systems least similar to our approach), progress to arm-worn mobile systems, and finally to mobile systems that do not require arm instrumentation (most

similar to our approach). We conclude this section with a review of systems that specifically used shadows for touch tracking.

We emphasize that although active-illumination shadow tracking has been used previously – including for the exact same use case as this present work – our particular instantiation offers a favorable mix of capabilities not demonstrated in any prior work (Table 1), including:

- Uses hardware already present in modern headsets (cameras and illuminators).
- Requires no instrumentation of the user's arms (i.e., bare hands).
- Enables ad hoc input for commonplace surfaces, including the user's skin.
- Works across lighting conditions, including in complete darkness.
- Works "out of the box", requiring no pre-registration or calibration of the surface, user, or environment.
- Offers high touch input accuracy (98.0% touch segmentation, 6.9 mm hover distance estimation mean error).

2.1 Detecting Touch with Instrumented Environments

The most straightforward way to add touch tracking to a surface is to instrument that surface directly. This is a well-explored area of research that we will not cover in depth for brevity. Some of the most common techniques have included placing microphones [16, 43, 46] or LIDAR [30, 56] on the surface, or cameras above the surface. For cameras, there have been many different optical approaches. For example, thermal cameras have been used to detect changes in heat left behind on the surface after touch [8, 23, 29, 31]. RGB cameras have been used to capture images of the fingernail and its color change during presses [2, 5, 34, 57] or even estimate fingertip pressure [5, 10, 11]. Perhaps the most popular technique has been to use fixed depth cameras operating above surfaces. After Benko and Wilson's works that pioneered this approach [3, 65], many other iterations have improved performance using flood-fill algorithms [68], combining infrared camera data [67], and applying machine learning [7]. Other lesser used optical approaches include multi-path interference from infrared depth cameras [49, 66], laser speckle imaging [45], and imaging reflections of the finger in mirrors and glossy surfaces [44, 70]. A major drawback of all these systems is that they need to instrument every potential surface or environment that the user may wish to interact with, which scales poorly. For this reason, research has looked into instrumenting the user with sensors to detect touches on ad-hoc surfaces, which we review next.

2.2 Detecting Touch with Finger/Hand/Arm-Mounted Sensors

When a user touches a surface, their touching finger produces a number of characteristic signals that could be used to detect touch contact. For instance, IMUs attached to the fingernail can detect spikes in deceleration that occur when a finger touches a surface [41, 42, 52], but are cumbersome for users to wear and recharge. More popular is an IMU ring form factor, which could be used to detect touch [12, 13] and mouse-like 2D inputs [24, 33, 50]. Beyond rings, researchers have also explored using IMUs placed on the wrist (like a smartwatch) to detect tap events [37], however this signal is less robust.

System Name	Sensor Hardware	Mobile vs. Stationary	Instrumented Hands / Arms	Supports Multitouch	World Input	Skin Input	Demonstrated in Darkness	Estimates Hover Distance	Touch Accuracy
TapLight [54]	IR Camera, Structured Light	Mobile	No	No	Yes	No	No	No	95.3%
OmniTouch [15]	Depth Camera	Mobile	No	Yes	Yes	Yes	No	No	96.5%
EgoPressure [76]	RGB Camera	Mobile	No	Yes	Yes	No	No	No	n.r.
MRTouch [69]	Depth & IR Camera	Mobile	No	No	Yes	No	No	No	96.5%
PressureVision++ [10]	RGB Camera	Stationary	No	Yes	Yes	No	No	No	89.3%
PlayAnywhere [64]	IR Camera, IR LED	Stationary	No	No	Yes	No	No	No	n.r.
Matsubara et al. [36]	IR Camera, IR LEDs	Stationary	No	No	Yes	No	No	No	96.1%
EgoTouch [38]	RGB Camera	Mobile	No	Yes	No	Yes	No	No	94.9%
Shadow Touch [32]	RGB Camera, White LED	Mobile	Yes	Yes	Yes	No	No	No	99.1%
EclipseTouch (Ours)	IR Camera, IR LEDs	Mobile	No	Yes	Yes	Yes	Yes	Yes (6.9 mm error)	98.0%

Table 1: Overview of key related work. Green is a positive attribute; red is negative. Refer to individual papers for study details.

Fingers also passively produce characteristic acoustic signals when tapping or swiping on a surface, which travel through the body and can be sensed with sensors mounted on the arm [9, 17, 27, 35]. Alternatively, touches can be sensed through changes in reflections of actively emitted acoustic signals. For instance, Mujibiya et al. [39], SoundTrak [71], and VersaTouch [51] emitted ultrasonic waves through transducers placed on the arm and finger to detect contact and pressure on the skin.

For detecting touches to the skin, one accurate approach that researchers have explored is detecting changes to RF signals transmitted through the user’s body. AtaTouch [26] detected subtle finger pinches by sensing changes in impedance between an antenna and the user’s body. SkinTrack [75] used two devices, a signal-emitting ring and a wristband receiver, with the body as an electrical waveguide to sense finger touch on the skin. ActiTouch [74] and ElectroRing [25] work in the same way, but move the transmitter and receiver to more convenient form factors. Finally, Z-Ring [62] used a single electrode ring that acted as a transceiver to sense bio-impedance changes in the hand. While these approaches are accurate, they are inherently limited to operation on conductive objects, such as the human body (but not most walls or furniture). A key limitation of all these prior systems is that they require instrumentation of the user’s finger, hand or arm, in addition to the XR headset.

2.3 Detecting Touch without Finger/Hand/Arm-Mounted Sensors

Rather than instrument a user’s hands with a device (likely special-purpose, as even smartwatches are rarely worn on the dominant hand), a more practical approach would be to have all necessary hardware contained within the XR headset/glasses. Dominant among these approaches is to use headset mounted cameras with computer vision to detect touches on surfaces.

As most modern headsets come with built-in 3D hand and world mesh tracking support, research has looked into inferring surface touch using this data. However, the world mesh modern headsets build is not currently accurate enough for touch detection purposes. For this reason, TriPad [6] required users to calibrate and define surfaces; hand tracking and dwells were used to instantiate touch planes and touches were detected by measuring fingertip proximity to the created plane. Richardson et al. [48] and Strelí et al. [55] focused on surface typing and used neural networks to analyze patterns of hand motion to detect surface taps with high accuracy.

However, both these systems required pre-registered surfaces to work, and could not detect stateful touches (including hover and touch-ups).

Another approach that has seen the most success in the past is using depth cameras. OmniTouch [15] and Imaginary Phone [14] were early among these efforts and used depth cameras to track the fingers and detect touches on the palm and other surfaces using a combination of flood-filling and contour detection. MRTouch [69] combined depth sensor data with infrared reflectivity data from a Microsoft HoloLens to improve touch detection accuracy. The problem with depth cameras, even today, is noisy signal – the difference between a finger touching vs. slightly hovering above a surface is hard to distinguish. For this reason, OmniTouch required users to lift fingers 20 mm above the surface to reliably separate hovering from touch events. This is awkward, and not like how one scrolls or types on their touchscreen devices. Moreover, depth cameras have other drawbacks including higher power consumption, lower resolution and lower framerates. Taplight [54] estimated fingertip depth and surface contact by using remote vibrometry (laser speckle sensing) obtained from a headset mounted laser and monochrome camera. This approach, while accurate in some contexts, does not function on many common surface materials, and can additionally fail from user head motion and multiple inputting fingers.

Most relevant to EclipseTouch is recent work on detecting surface contact using only headset cameras. This is challenging, as the finger directly occludes the point of touch contact, and the system must adapt to surface materials, lighting conditions, touch types, and skin tones. PressureVision [11] and later PressureVision++ [10] described deep learning-based approaches to detecting contact pressure of the hand to a surface, albeit with a fixed desk-mounted camera and restricted lighting and surface conditions. EgoPressure [76] extended this work to headset-mounted cameras by collecting a large egocentric hand pressure estimation dataset, but their system ran offline (not in real-time) and required ambient illumination. In our prior work EgoTouch [38], we demonstrated a system for detecting on-skin touch and force using only headset RGB cameras, by looking at patterns of skin deformation, color change and shadow convergence. Similar to EgoTouch, PalmPad [18] could also detect touches to the skin with a headset-mounted RGB camera, however it could not estimate force and only supported the palm. Both EgoTouch and PalmPad ran in real-time, but only worked on the skin, required ambient illumination (i.e. did not work in the dark), and were susceptible to false positives from extraneous shadows in the environment.

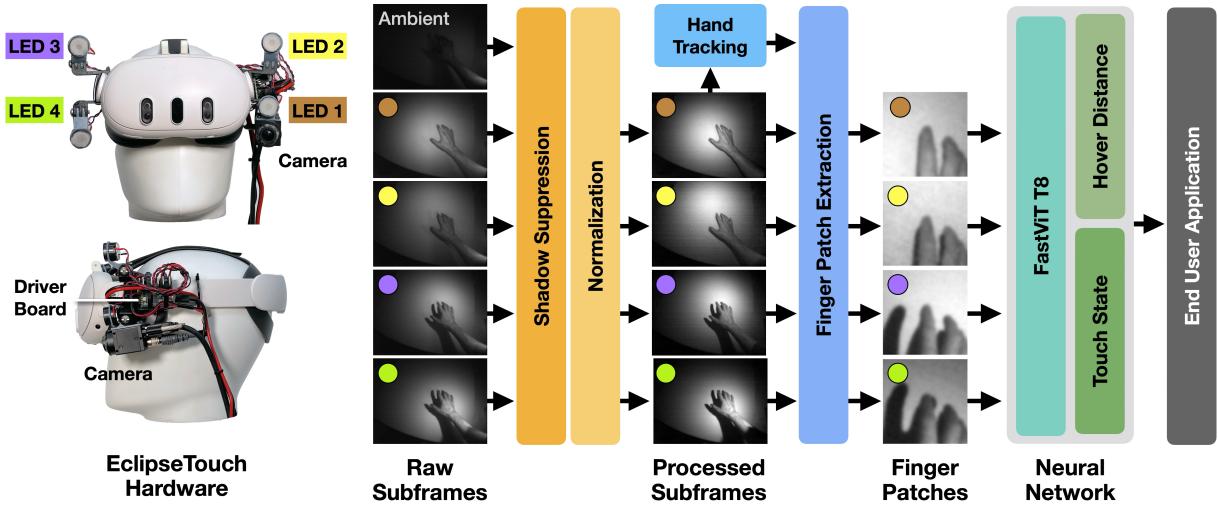


Figure 2: High-level overview of EclipseTouch’s experiment hardware and software pipeline.

The fundamental problem with all these prior approaches is that they rely on existing sources of illumination. However, ambient light is uncontrolled – it can be diffused, harsh, oblique, bright, multicolored, single/multi-point, and even non-existent (dark). With EclipseTouch, we sought to overcome these limitations by controlling the shadows cast by the finger on the surface.

2.4 Detecting Touch using Shadows

To conclude our literature review, we now specifically discuss systems that have leveraged shadows to track user interactions, as this most closely relates to our technical approach.

Starting with seminal work, we have Myron Krueger’s VideoPlace [28], which utilized user silhouettes for interactivity (though not true shadows). As far as we are aware, the earliest known work to leverage “shadow shape analysis” at the fingers for *touch input* is Andy Wilson’s PlayAnywhere [64]. This system used a camera and illuminator that was fixed with respect to the input plane by virtue of the system being placed on a table. This system laid the conceptual groundwork for using shadows for detecting touch, measuring the decreasing distance between the shadow and fingertip.

More recent touch systems employing fixed cameras and fixed illuminators include ShadowReaching [53], Iacolina et al. [22], and Thomas [58]. Matsubara et al. [36] and Niikura et al. [40] used the same system, which featured two fixed infrared illuminators and a fixed camera capturing image pairs (one for each active illuminator). More ambitious is to have only a fixed camera and rely on existing natural or artificial light sources (i.e., no special illuminators). Adajania et al. [1], Paper Piano [60], ShadowSense [21], and Posner et al. [47] use this approach. However, with no control of the positioning between the camera and light sources (one or many, harsh or diffuse, bright or dim, etc.), input tends to be brittle, working well in some cases and failing in others.

Most similar to EclipseTouch, is ShadowTouch [32]. Unlike the above prior work, ShadowTouch is worn and mobile (i.e., not reliant on fixed external infrastructure). Like EclipseTouch, the system used an egocentric headset camera. Unlike EclipseTouch, ShadowTouch

requires a special-purpose LED wristband. As there is no synchronization between the LED and camera, the LED is persistently lit, which is prohibitively energy consumptive for a wearable. Additionally, the authors note that they do “not have a special design to alleviate the ambient light interference” [32], an important part of our pipeline. We also move beyond ShadowTouch in terms of evaluation generalizability. In ShadowTouch’s study, only three light-colored and matte surfaces are tested, only in a horizontal setting, and in one typically-lit environment. In this work, we test 12 surface materials (Figure 6), both dark and light, and matte and reflective. We also test vertical and horizontal orientations, and across three lighting conditions (typical lighting, bright, and dark). We note that our evaluation results show (Section 5) both systems to be of comparable accuracy; in other words, we achieve the same accuracy without the need for a special wearable.

3 Implementation

As a prototype platform, we instrumented a Meta Quest 3. In the following subsections, we step through the various hardware and software components that make up EclipseTouch.

3.1 Camera

We affixed a HT-SUA33GM-T1V-C USB 3.0 camera [61] to the bottom left of our prototype headset (Figure 2). This global shutter camera has a modest resolution of 640x480 pixels, but is sufficient for our needs (i.e., our shadows are large and textureless, not requiring high detail). We fitted the camera with an 850 nm bandpass filter, matching the wavelength of our LED illuminators (discussed next). Even with this filter, other light sources can cast visible shadows at 850 nm, most notably the sun, as well as incandescent and halogen lights. To account for this, our pipeline includes an extraneous shadow suppression process, described in Section 3.5. As we rely on only a narrow frequency band of light, we purposely selected a monochrome camera with a wideband CMOS sensor (with no

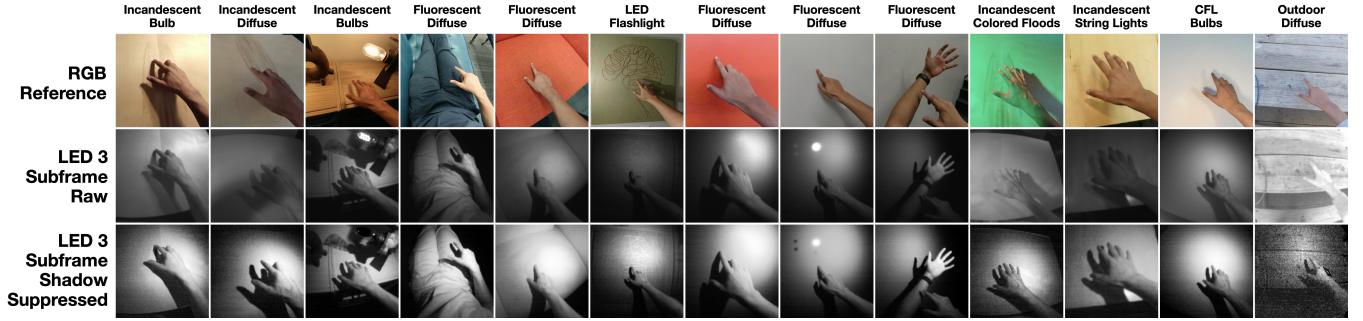


Figure 3: Examples of ambient shadow suppression output across a variety of materials, lighting types (incandescent, fluorescent tubes, CFL, LED, sun), and illumination conditions (diffuse, floodlight, point source). Note that shadows visible in visible light (RGB Reference) may be invisible in infrared and vice-versa. Figure 4 provides a step-by-step example of shadow suppression.

Bayer pattern, which would preclude imaging infrared light). Finally, this camera features an external hardware trigger, which we use to synchronize with our LED illuminators. The camera streams video over USB to a laptop, where our software runs. The maximum framerate of the camera is 791 FPS, though we operate it about half this speed to increase exposure time.

3.2 Illumination Geometry

As already discussed, EclipseTouch relies on active illumination to create structured shadows. We use LEDs with a wavelength of 850 nm (infrared), that are safe and invisible to humans, even in darkness. The LEDs are rated at 3 W, but we drive them at 1.1 W as a power conservation measure. Importantly, our prototype was built to serve as a vehicle for investigation. As such, we over-provisioned our headset with LEDs in each of the four corners. For the corner with the camera, the LED is placed directly above the camera. This arrangement can be seen in Figure 2. As we will evaluate and discuss later, some illuminator locations are more valuable than others, and so a commercial implementation would use fewer illuminators, potentially just one (as can be seen with our final prototype in Figure 14).

3.3 Driver Board

We use a Teensy 3.2 microcontroller and custom MOSFET LED driver board to precisely control the timing of our LED illuminators and camera frames (microsecond precision). Our Teensy firmware uses the following five-step LED "firing sequence": No LEDs on, LED 1 on, LED 2 on, LED 3 on, and LED 4 on. Each step in the sequence has a duration of 2.5 ms, and at each step the camera is triggered using its external pinouts. The firing sequence loops continuously, producing a 400 FPS raw video stream.

Our camera uses an exposure time of 2.4 ms. We note that this is a comparatively short exposure time for a camera with a small sensor (1/5.6"), but this is not an issue for EclipseTouch because we are actively illuminating the scene. Furthermore, human skin is reflective in infrared, and the user's hands are never more than 70 cm away from the headset. As can be seen in Figures 3 and 6, the hand is readily seen, and the shadows cast are crisp and dark.

3.4 Video Stream

On our laptop receiving the 400 FPS camera stream, we read five frames at a time (i.e., a complete firing sequence), and composite this data into a new, singular frame containing multiple illumination sources: five 640x480 images side-by-side in a row. Thus, this new composited stream has a framerate of 80 FPS. The time between the start of the first frame (no illumination) and the end of the last frame (LED 4 on) is approximately 12.5 ms. This short duration means that even when the hands are in motion, the image can be stacked for image processing as though they were taken at essentially the same moment in time.

3.5 Extraneous Shadow Suppression

In addition to our infrared LEDs, other light sources with 850 nm wavelengths will cast finger shadows. Notable light sources include the sun, as well as some artificial lights, including incandescent and halogen bulbs (see example shadows in Figure 3). These shadows can generate false events, and so it is desirable to filter them out.

Importantly, light intensity is additive on image sensors (i.e., each CMOS pixel measures accumulated light, and if two or more light sources are contributing photons to this pixel, it will simply be the sum of intensities). We can use this property to great effect for removing unwanted shadows. Specifically, we can take our "no LEDs on" subframe, which captures any shadows generated from extraneous light sources, and simply subtract this from our other four subframes. This has the effect of removing the contribution of ambient light on those subframes, leaving only the illumination from that specific subframe's LED. An illustration of this process is shown in Figure 4, with example outputs across various environmental conditions seen in Figure 3.

3.6 Finger Tracking

With extraneous shadows removed, we next move to track the hands in front of the user. In an integrated system, this information would already be available from the hand tracking provided by the headset software. However, as we are using our own camera, we cannot simply use the Quest 3's hand tracking result, and instead must compute our own. For this, we utilize our LED 1 subframe, which provides an illuminated view of the hand with minimal shadows (as LED 1 is almost directly inline with the camera). We

run Google’s MediaPipe hand tracker [72], which provides 21 2.5D hand keypoints, though we only use the five fingertip points. We create 64x64 pixel patches centered on each fingertip. We use the wrist and Metacarpophalangeal (MCP) hand joints to normalize the size of the fingers (i.e., scale, irrespective of distance from the camera) scale, and then the MCP and Proximal interphalangeal (PIP) finger joints to normalize the rotation (so that all fingers are pointing upwards in the patch). These finger patches are then passed to our ML model, described next.

3.7 Machine Learning

Our deep learning model takes in as input a finger patch and a finger ID and jointly predicts touch state and hover distance. Our model is a hybrid vision transformer, built on top of the FastViT T8 [59] backbone. Finger patches contain N channels ($N \in 1, 2, 3, 4$), one for each of the illuminator subframes. Finger patches are first passed through the backbone to produce image embeddings of size 768. These embeddings are then concatenated with a 5 dimensional one-hot encoded vector of the finger ID (thumb=1, pinky=5) to produce an embedding of size 773. Finally, this embedding is passed through a multi-layer perceptron (2 layers, hidden dimension 128,

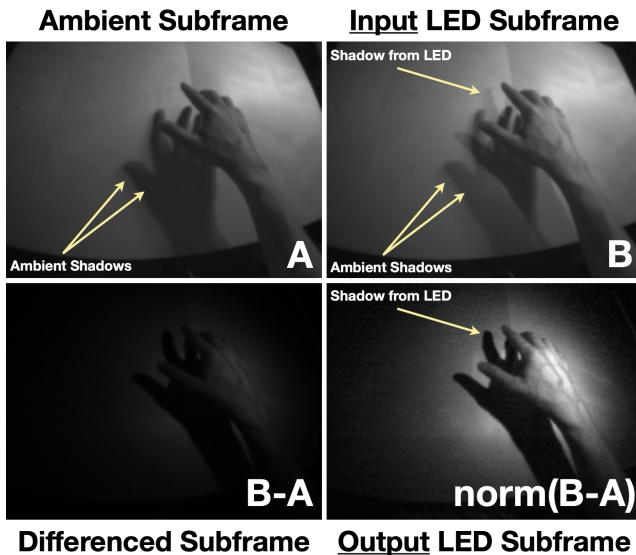


Figure 4: Overview of our shadow suppression process. Our ambient subframe (A) captures shadows cast by ambient light sources. When an illuminator is on (LED 3 in this example), we cast a new additional shadow into the scene (B). Note this shadow can be weaker than ambient shadows, as is the case in this example (see also Figure 3). We then subtract the ambient frame from the LED-illuminated subframe (B-A). Having now subtracted ambient light, the frame becomes darker, and represents light only cast by the headset LED. To compensate for variable scene brightness (e.g., varying surface albedo and hand distance) we perform a final normalization — $\text{norm}(B-A)$ — which accentuates the shadow. Note only a single shadow remains, the one cast by the headset LED.

GeLU [20] activations) that encodes the scaled distance of the finger to the surface. Touch state is obtained by sigmoid activating and thresholding this value. We smooth touch and hover distance predictions with a mean filter over the 30 most recent frames.

3.8 Model Training Protocol

In our subsequent user study, we employ a leave-one-participant-out cross validation scheme to train and evaluate our models. Our models were trained using the PyTorch and PyTorch Lightning deep learning frameworks. We initialized the FastViT backbone with ImageNet pretrained weights. We first trained the model to estimate touch state (by minimizing binary cross-entropy loss) and later fine-tuned the model to encode distance in the final logit (by minimizing the sum of mean absolute error and mean squared error). Models were trained for 10 epochs using the Adam optimizer, a batch size of 128 and a learning rate of 0.000003, which took about two hours on an NVIDIA 2080 Ti GPU.

3.9 Compute and Power Consumption

We carefully designed our model to be able to run as a lightweight background process, concurrent with numerous other models that already run on XR headsets (hand/body tracking, SLAM, etc). After training, we re-parameterize the model [59] to an equivalent one with fewer parameters (total 3.3M parameters). On an M2 Macbook Air — with similar hardware to the Apple Vision Pro — our model has an inference time of 0.47 ms. This means that our model can potentially run at ~2000 FPS, or run at e.g., 60 FPS consuming a small fraction of the headset’s processing power.

As noted previously, our infrared LED illuminators consume 1.1 W when active. Only one LED is active at a time, and no LEDs are active 1/5th of the time, yielding a mean power draw of 0.9 W for illumination. Our microcontroller and LED driver board consumes 0.3 W. Our camera, running at 400 FPS, draws 0.8 W. As one reference point, the Meta Quest 3 draws ~8.6 W of power during use (giving its 18.9 Wh battery a stated 2.2 hours of runtime).

3.10 Compatibility with Existing Headsets

We note that several popular XR headsets already contain the requisite hardware to enable EclipseTouch. For example, the Apple Vision



Figure 5: XR headsets on the market already include integrated infrared illuminators and infrared-sensitive cameras, suggesting EclipseTouch could be enabled via a software update. Additionally, we note that future AR glasses could also incorporate a single camera and LED in opposite corners of the frame, much like Ray-Ban Meta AI Glasses do today.

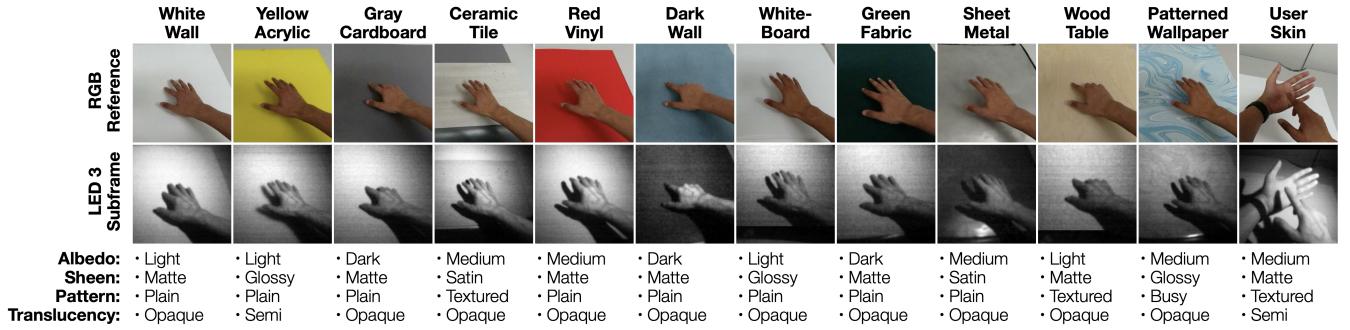


Figure 6: The twelve materials we tested in our study. Note the difference in appearance in RGB vs. infrared color spaces.

Pro contains two infrared illuminators and six monochromatic cameras sensitive to infrared light (Figure 5, left). The low-cost Meta Quest 3S contains two infrared illuminators and two monochromatic cameras (Figure 5, right). In both cases, the illuminators are used to boost hand tracking performance in low-light conditions [19]. However, their arrangement is also perfect for EclipseTouch—two infrared illuminators offset from one or more infrared cameras (more discussion in Section 5.1). Thus, it is likely that EclipseTouch could be enabled with a software update (at present, neither headset provides ad hoc surface touch segmentation, other than in a rough way using the native hand tracking).

4 Data Collection Protocol

To collect data to train and evaluate EclipseTouch, we recruited 10 participants (4 female, 6 male, all right-handed) for a one-hour user study. Participants were compensated \$20 for their time. After completing consent paperwork, participants were fitted with our EclipseTouch-instrumented Quest 3, which streamed our 80 FPS video composite to a local desktop via USB where it was saved. The study was conducted in a windowless room with controlled lighting. We measured lux, reported later, using a Sper Scientific 840022 Light Meter. During the study, participants stood in front of a standing desk or a wall. Unlike prior work, we did not restrict participants head motion or distance to surfaces.

We designed our study to capture a variety of conditions such that we could later analyze system performance across different materials ($n=12$, of varying albedo, sheen, patterns and translucency; Figure 6), surface orientations (horizontal and vertical), and lighting conditions (bright, typical, and dark; Figure 7). It was not possible to fully cross these conditions due to combinatorial explosion, and so we designed blocks of sessions to collect data, varying a single experimental factor at time.

All of the individual sessions followed the same basic data collection procedure. First, participants were instructed to continuously touch and drag on a presented surface with their index finger, during which time 30 seconds of data was recorded. This was immediately followed by a second session capturing 30 seconds of data in which participants were asked to hover their finger above the surface and to perform repeated in-air taps. These trials provided positive and negative examples to train our machine learning model.

To collect data across a variety of surfaces, we curated a set of 11 diverse materials — including wood, plastic, metal, fabric, and

painted/printed surfaces — seen in Figure 6 (along with a summary of their properties). These material samples were all cut down to 60×40 cm so as to be easily swapped and organized during the study. Both touch and hover sessions of data were collected for all 11 materials, in a random presentation order, in a horizontal orientation (placed on a desk), and in typical home lighting (measured at 127 lux, with typical home illumination around 100-200 lux [63]).

Next, we collected data in two additional lighting conditions: bright (1633 lux) and dark (0.005 lux). For reference, typical TV studio lighting is around 1000 lux and a quarter moon on a cloudless night is 0.01 lux [63]. For this study block, we reduced our material set down to two: white wall and patterned wallpaper. The former was a representative plain and light-colored material, and the latter served as a more challenging surface, being darker, glossy, and

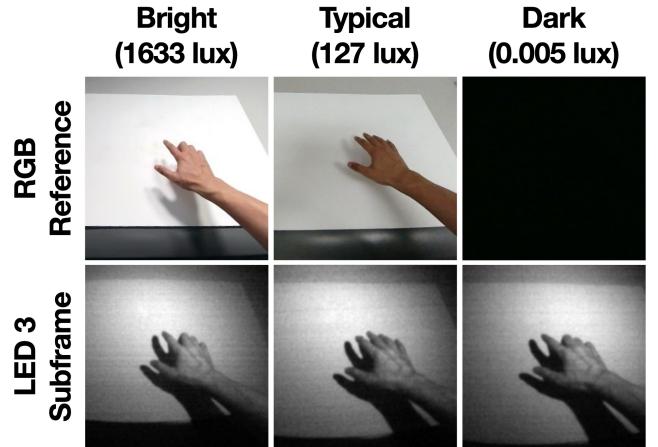


Figure 7: We tested three lighting conditions in our study: bright, typical, and dark. For reference only, we show the scene as it appears to a standard RGB camera along the top row. The bottom row shows the LED 3 illuminated subframe at the end of our pipeline (not shown are LED 1, 2 and 4 subframes, but the results are equivalent). In short, EclipseTouch is reasonably agnostic to ambient lighting condition as it provides its own illumination. Note also how shadows cast from ambient sources are not visible in the processed frame due to shadow suppression.

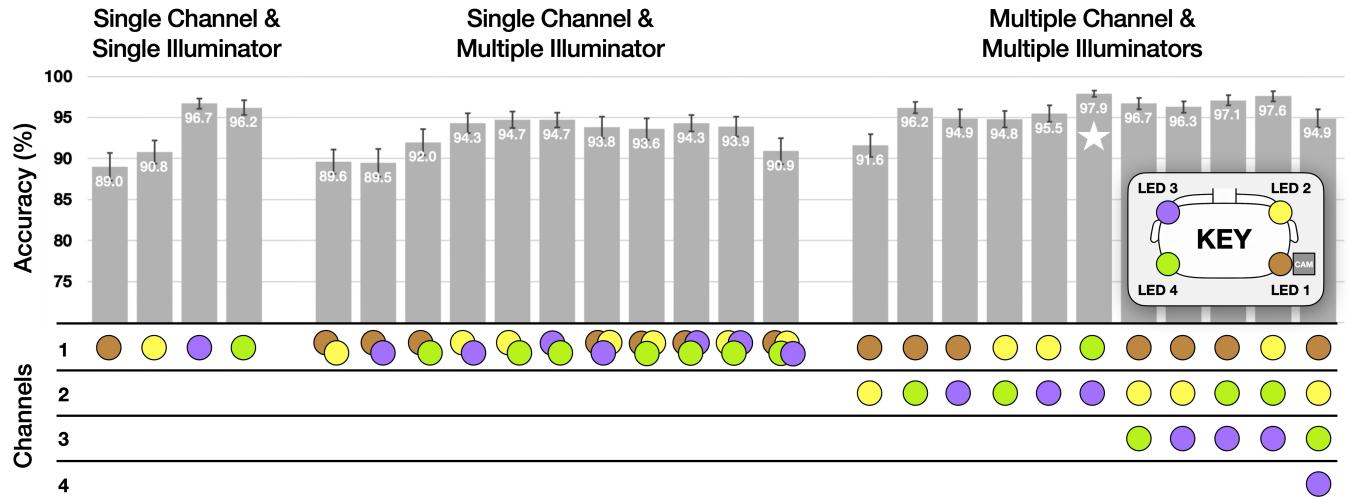


Figure 8: Results across illuminator combinations. A colored dot indicates that the corresponding illuminator is active in that channel. The best combination (LED 3 & LED 4, operating in separate channels) is marked with a star.

patterned. For these two materials, we collected two additional sessions of data in each lighting condition and in a horizontal orientation. Note in the previous block of sessions, we already captured data for these two materials under typical lighting (127 lux). For our three lighting conditions, we provide reference visible-light photos and infrared-illuminated subframes in Figure 7. We also collected data for surfaces in a vertical orientation. For this, we use our same pared-down material set — white wall and patterned wallpaper — and our typical home lighting condition.

Finally, the user’s skin is a special, high-value input surface with unique optical properties (most notably subsurface scattering, creating diffuse shadows). In order to evaluate EclipseTouch’s ability to extend to skin input (our 12th material), we collected eight sessions of data on the palm: two sessions each with the hand held horizontal in our bright/typical/dark lighting conditions, and two sessions with the hand held vertically in typical lighting.

In total, each participant completed 42 sessions of data collection. With each session producing 30 seconds of data and a system framerate of 80 Hz, multiplied by 10 participants, this process yielded 1,008,000 individual frames for training and analysis.

5 Results and Discussion

Our system prototype and study procedure was purposely designed to enable investigation of several important factors. First and foremost, we ran an ablation study to identify the most promising illuminator geometry, balancing accuracy with practicality. All subsequent results use this arrangement. Input to the skin is broken out as a special discussion. We conclude the section with two supplemental studies: hover distance estimation and multitouch. For touch prediction, we report classification accuracy and for hover distance estimation, we report mean absolute error in millimeters. Other than our illumination geometry ablation study (the next section), all results reported in this section are trained and tested with leave-one-participant-out cross validation, with no user or surface calibration. In all experiments, we always test on unseen users.

Across all materials (including skin), in both orientations and all three lighting conditions, EclipseTouch achieves an overall mean accuracy of 98.0% (SD=0.3%) using its best-performing LED 3 & 4 illumination geometry.

5.1 Across Illumination Geometries

First, we evaluate the performance of EclipseTouch under different illuminator configurations. Our prototype hardware has four illuminators (Figure 2), which leads to 15 possible combinations ($2^4 - 1$) for illuminator placement on the headset. To enable triggerless operation and higher framerates, illuminators could also be turned on in groups of two, three, or four, and so we additionally ablate these 11 extra combinations. Due to the additive property of light transport, we can simulate these combinations by averaging the subframes from each of the LEDs. Figure 8 shows all the 26 illuminator combinations we tested. Prior to training each model, we modify the model architecture to accept images with different numbers of channels, by modifying the input channels of the first convolution layer of the backbone. We used six participants’ data for training, and the remaining four participants’ data for testing.

Results from this ablation study are shown in Figure 8. Focusing first on the Single Channel, Single Illuminator results, it is clear that LED illuminators 3 and 4 yield the best results (96.7% and 96.2% accuracy). Indeed, there is a clear pattern in performance relating to illuminator-camera distance. Figure 2 shows shadows formed by each of these illuminators. We note that illuminators 3 and 4 are the furthest offset from the camera and produce the most prominent shadows of the touching finger. Next farthest from the camera is illuminator 2, which produces notably smaller shadows of the touching finger. Finally illuminator 1, which is in-line with the camera, produces almost no shadow, leading to the worst performance of all the illuminators (89.0%).

Moving on to Single Channel, Multiple Illuminator configurations, where multiple illuminators are on at the same time, we observe that the best combinations are LED 3 + 4 (94.7%), as well as

LED 2 + 4 (also 94.7%). While this performance is high, we observe that all combinations in this category consistently perform worse than their individual channel counterparts, as well as the best performing single-illuminator configurations. Simultaneously turning on multiple illuminators had no performance benefit over using one or more LEDs at optimal locations.

Finally, looking at Multiple Channel, Multiple Illuminator results, we find the best-performing combination is LED 3 & 4 at 97.9%, followed closely by LED 2, 3 & 4 at 97.6%. This makes sense, especially considering the fact that LED illuminators 3 and 4 already achieved high accuracies on their own.

Notably, the accuracy with even just one LED at an optimal location is similar to the best performing combination (96.7% for LED 3 alone vs. 97.9% for LEDs 3 & 4). This is promising, as existing headsets already include a pair of illuminators offset from integrated cameras (see Figure 5 and Section 3.10), potentially allowing EclipseTouch to be enabled through a software update. We also created a final prototype for demo purposes, seen in Figure 14, which features a single illuminator based on these results. For all subsequent study results, we report performance on this best-performing, multi-channel illuminator configuration: LED 3 & 4.

5.2 Across Surface Materials

A key component of our study was to evaluate the performance of EclipseTouch on a variety of everyday materials (see Figure 6 for the 11 non-skin materials that we tested). Importantly, these materials had a wide range of properties, including varying albedo, sheen, patterns and translucency. Results from this evaluation are shown in Figure 9. Across all materials tested, performance remains high, averaging 98.6% accuracy. This ranges from 96.6% for the yellow acrylic material to 99.8% for the green fabric material. Note that 9 out of 11 materials tested have accuracies above 98.0%, indicating strong generalization of EclipseTouch across surface materials.

Performance remains similar across albedo types, with light materials averaging 98.2% (SD=1.3%), medium materials averaging 99.0% (SD=0.7%), and dark materials averaging 98.4% (SD=1.4%) accuracy. This is encouraging, since darker materials often pose a challenge to other vision-based systems.

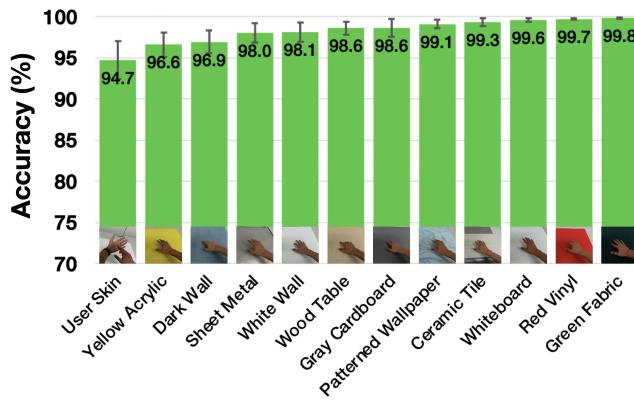


Figure 9: Touch classification accuracy vs. surface material.

Similarly, performance remains comparable across sheen types, with matte materials averaging 98.6% (SD=1.1%), satin materials averaging 98.7% (SD=0.9%), and glossy materials averaging 98.4% (SD=1.6%). Note that while we do include glossy materials, none of our materials were fully reflective, like glass. These materials do not produce shadows and thus cannot be supported.

Also encouraging is that performance remains high with materials with busy patterns; over 99.0% accuracy for our patterned wallpaper, wood table, and ceramic tile surfaces.

The worst performing material was the yellow acrylic, which had a mean accuracy of 96.6%. This material is semi-translucent, and produces shadows with softer edges as compared to the other materials in our study, potentially reducing performance. Opaque materials averaged 98.8% accuracy (SD=0.9%).

5.3 Across Lighting Conditions

Next, we analyze performance across lighting conditions. Figure 7 shows the three lighting conditions we tested in this study, ranging from dark (0.005 lux) to bright (1633 lux) lighting. Results for the three materials we studied across lighting conditions can be seen in Figure 10. On average, touch accuracies were 98.1% (SD=1.5%) for the bright condition, 97.3% (SD=2.3%) for the typical condition, and 99.0% (SD=0.8%) for the dark condition. Overall, performance remains similar across lighting conditions, with a slight increase in performance in the darkest lighting condition. This is an encouraging result, since most prior work either did not test or did not work well across such a wide range of lighting conditions.

5.4 Across Surface Orientations

Touch performance across the two orientations we tested can be seen in Figure 10. On average, performance was slightly better for the horizontal condition (97.3% accuracy, SD=2.3%) vs. the vertical condition (95.8% accuracy, SD=2.4%). Notably, the patterned wallpaper material had a sharp decrease in performance from 99.1% to 93.5% accuracy. We note that in our user study, participants typically had their hand closer to the headset in the vertical condition vs. the horizontal condition, which could have contributed to this decrease in performance. Furthermore, shadows of the touching finger look different in the vertical condition as compared to the horizontal condition, and since the majority of our training data was collected in the horizontal orientation, we hypothesize our models may be slightly biased towards this orientation.

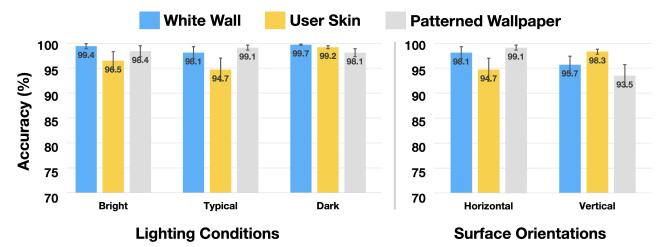


Figure 10: Touch classification accuracy vs. lighting condition (left) and surface orientation (right).

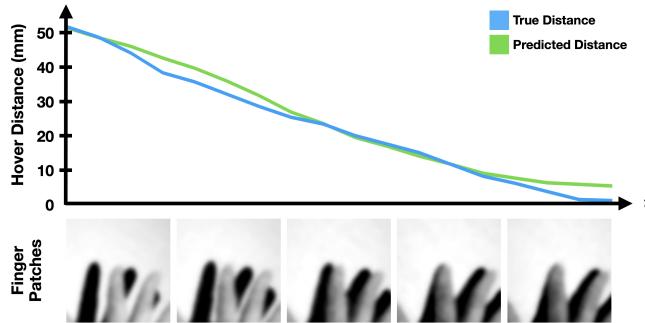


Figure 11: Example 250 ms sequence of an index finger descending to tap a surface. Note how the shadow evolves over time (illuminated by LED 3), converging towards the finger, and essentially disappearing upon contact. Our model uses this visual information to predict hover distance.

5.5 On-Skin Touch Detection

An important surface material to study for touch input is the user’s skin, as it is always available for input on the go. However, skin has unique properties that set it apart from the other materials we tested. It has varying albedo, surface texture, and some translucency (leading to subsurface scattering of light). It is also deformable and non-planar. For this reason, we separately trained a model for skin input. We envision this model being used when the XR headset’s existing hand tracking detects probable hand-to-skin input.

On-skin touch detection results are shown in Figures 9 and 10. Compared to other materials, detection accuracy was slightly lower, at 94.7% accuracy ($SD=2.3\%$). We note, however, that this performance is still competitive with prior work [15, 38]. Across lighting conditions, performance also remains high, with a noticeable jump in performance in the dark (99.2% accuracy). We note that prior vision-based on-skin touch systems, such as our own EgoTouch [38], has essentially 0% accuracy in the dark. Across orientations, performance is higher in the vertical orientation (98.3%) vs. the horizontal orientation (94.7%).

5.6 Supplemental Study: Hover Distance

As a user moves their finger towards a surface, the shadow cast by that finger moves and the distance between the fingertip and the tip of the shadow changes proportionally (Figure 11). For this reason, we hypothesized that it would be possible to train a model to estimate finger hover distance above a surface. To train and evaluate this model, we ran a small supplemental user study with 5 participants (2 male, 3 female). Participants filled out consent paperwork and were fitted with the EclipseTouch prototype. Participants stood in front of a table in a typically lit room. To collect ground truth distance of the finger from the surface, we affixed a HD USB webcam to the side of the table, so as to track the participant’s touching finger from the side. At the start of the study, the experimenter calibrated the pixel displacements of the user’s hand in the webcam view to real world units (mm). Then, participants were asked to lift their index finger up and down above a certain location on the surface. Participants were instructed to only lift their finger vertically, so as to accurately track their fingertip’s ground truth

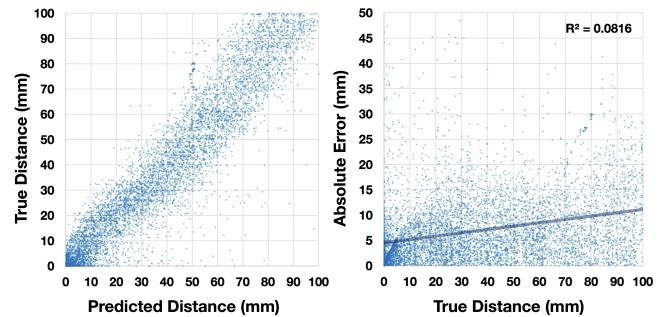


Figure 12: Left: Predicted vs. true hover distance. Overall, EclipseTouch is fairly linear and accurate, up to about 10 cm (mean error of 6.9 mm). Right: Mean absolute hover distance error vs. true distance – note the slight upward trend in error.

distance from the surface. We collected three sessions lasting 60 seconds each, for a total of 72,000 frames at 80 FPS.

To train this model, we simply fine-tuned our existing EclipseTouch model to directly estimate hover distance in the output touch classification logit (before activation). The main motivation for this design was that hover distance and touch classification confidence are highly correlated. Furthermore, this allowed us to have a single unified model that estimated both hover distance and touch state simultaneously. All models were trained with a leave-one-participant-out cross validation scheme.

Results from this supplemental study can be seen in Figure 12. In our hover distance region of interest, below 10 cm, our model had a mean absolute error of 6.9 mm. For ground truth distances less than 1 cm away from the surface, our model had a mean absolute error of 2.5 mm. Furthermore, we note that error increases slightly as distance from the surface increases (Figure 12). Example predictions of this model as a user touches a surface can be seen in Figure 11.

5.7 Supplemental Study: Multitouch

To evaluate the performance of EclipseTouch on multiple fingers, we ran a small supplemental study (in tandem with the hover distance estimation study). For this study, we collected four types of sessions, all with the white wall material in typical lighting. First, participants were asked to touch and drag across the surface with all their fingers simultaneously. Next, participants were asked to hover close to the

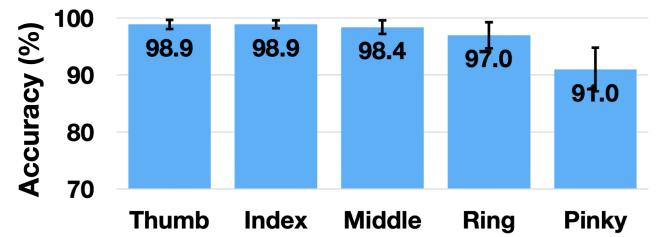


Figure 13: Results from our multitouch supplementary study, broken out by inputting finger.



Figure 14: Based on our evaluation results, we created a final prototype featuring a single infrared emitter and camera.

surfaces and to perform in-air taps with all their fingers. Third, participants were asked to perform a pinch and zoom motion with their index and thumb fingers while touching the surface. Finally, participants were asked to perform the pinch and zoom motion in the air while hovering close to the surface. Each session lasted 30 seconds, and we collected three rounds per session, for a total of 144,000 frames at 80 FPS. We then fine-tuned our models with this data. Similar to before, we employed a leave-one-participant-out cross validation scheme to train our models.

Results from this study are shown in Figure 13. Similar to [32], we observe that the model performs similarly across the thumb, index and middle fingers (98.7% accuracy, SD=0.3%). Performance drops slightly for the ring finger (97%) and drops further for the pinky finger (91%), as they are sometimes not visible to the camera.

6 Limitations & Future Work

While EclipseTouch represents a useful and practical increment over prior work, there is still room for improvement to achieve touchscreen-like performance. Foremost, similar to other vision-based approaches, EclipseTouch needs line-of-sight to the touching finger to function. On modern headsets, this issue has been partially addressed by incorporating multiple cameras on the headset, expanding the field of view outside that of the users.

We also note that EclipseTouch does not work across all surface materials. In particular, highly-reflective and transparent materials will fail, such as mirrors and glass (as no shadows are visible). In future work, the reflection of the finger on the surface itself could be used to estimate touch contact [44]. Apart from this, we also observed that very dark materials (in infrared) did not cast shadows, nor did highly 3D-textured surfaces (e.g., piled carpets, fur, fluffy clothing). Across all materials tested in our study, the two worst-performing were User Skin and Yellow Acrylic (the only two semi-translucent materials we tested). One hypothesis for this reduced performance is that subsurface scattering interferes with cast shadows. In the future, EclipseTouch could use its cameras and computer vision to identify unsuitable surfaces, and steer users to utilize compatible ones.

The power draw of EclipseTouch could be an obstacle to adoption in commercial systems (read more about the power consumption of our prototype in Section 3.9). In general, active illumination

is power expensive. In systems using light emitters, such as the LIDAR and TrueDepth sensors in Apple’s iPhones, the sensor is only turned on when needed. For example, the auto-unlock feature is first triggered by a motion event detected by a lower-power IMU, before turning on the more power-expensive TrueDepth sensor for Face ID. Even when on, the duty cycle is kept very low. A similar approach could be used for EclipseTouch. Contemporary XR headsets already build a model of the environment and track the user’s hands for input. This data could be used to activate EclipseTouch opportunistically, when the hands are closer than e.g., 30 cm to a surface. When active, the infrared LEDs could be strobed for very short durations (our cameras are already externally triggered, so synchronization is not an issue), further reducing power consumption.

Another condition where EclipseTouch does not presently function is in direct sunlight. As noted in our study, our bright lighting condition was 1633 lux (exceeding that of typical TV studio lighting at 1000 lux [63]). It was certainly bright compared to a typical office, but not bright compared to the direct Sun, which can be 100,000 lux [63]. At this level of illumination, our 1.1 W LEDs cannot compete, and the shadows they cast simply disappear into noise. Other active illumination sensing methods have employed various strategies to combat this, including modulating and polarizing light, as well as using very narrow bandpass optical filters. This is how, e.g., depth sensors such as Microsoft’s Kinect and Apple’s iPhone LIDAR can work in outdoor scenes.

There are also opportunities to further generalize EclipseTouch to new surfaces. For instance, we are exploring how game engines could be used to generate synthetic data, since the phenomenon of shadow casting is well developed for games. Alternatively, we could also use new foundation models for relighting [73] and hand generation [4] to augment and create new synthetic data on a variety of materials. Future work could also explore touch detection on irregular surfaces, beyond the palm, as well as fusing multiple vision modalities (e.g. RGB & IR).

7 Conclusion

We have presented EclipseTouch, a new headset-only system for detecting touches on everyday surfaces, including the user’s skin, using worn infrared shadow casting. Moving beyond prior work, our results show that this approach is quite accurate (98.0% touch accuracy, 6.9 mm hover distance error), and works across a wide range of surface materials, lighting conditions and orientations, while running efficiently and with low latency. Our ablation studies reveal optimal illuminator arrangement geometries and suggest that EclipseTouch could be implemented in existing headsets through a software update.

References

- [1] Y. Adajania, J. Gosalia, A. Kanade, H. Mehta, and N. Shekokar. 2010. Virtual Keyboard Using Shadow Analysis. In *2010 3rd International Conference on Emerging Trends in Engineering and Technology*. 163–165. <https://doi.org/10.1109/ICETET.2010.115> ISSN: 2157-0485.
- [2] Ankur Agarwal, Shahram Izadi, Manmohan Chandraker, and Andrew Blake. 2007. High precision multi-touch sensing on surfaces using overhead cameras. In *Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP’07)*. IEEE, 197–200.
- [3] Hrvoje Benko and Andrew Wilson. 2009. DepthTouch: Using depth-sensing camera to enable freehand interactions on and above the interactive surface. In

- Proceedings of the IEEE workshop on tabletops and interactive surfaces*, Vol. 8. 21.
- [4] Kefan Chen, Chaerin Min, Linguang Zhang, Shreyas Hampali, Cem Keskin, and Srinath Sridhar. 2025. FoundHand: Large-Scale Domain-Specific Learning for Controllable Hand Image Generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*. 17448–17460.
 - [5] Nutan Chen, Göran Westling, Benoni B. Edin, and Patrick Van Der Smagt. 2020. Estimating Fingertip Forces, Torques, and Local Curvatures from Fingernail Images. *Robotica* 38, 7 (July 2020), 1242–1262. <https://doi.org/10.1017/S0263574719001383>
 - [6] Camille Dupré, Caroline Appert, Stéphanie Rey, Houssem Saidi, and Emmanuel Pietriga. 2024. TriPad: Touch Input in AR on Ordinary Surfaces with Hand Tracking Only. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. ACM, Honolulu HI USA, 1–18. <https://doi.org/10.1145/3613904.3642323>
 - [7] Neil Xu Fan and Robert Xiao. 2022. Reducing the Latency of Touch Tracking on Ad-hoc Surfaces. *Proc. ACM Hum.-Comput. Interact.* 6, ISS (Nov. 2022), 577:489–577:499. <https://doi.org/10.1145/3567730>
 - [8] Markus Funk, Stefan Schneegass, Michael Behringer, Niels Henze, and Albrecht Schmidt. 2015. An Interactive Curtain for Media Usage in the Shower. In *Proceedings of the 4th International Symposium on Pervasive Displays*. ACM, Saarbruecken Germany, 225–231. <https://doi.org/10.1145/2757710.2757713>
 - [9] Jun Gong, Aakar Gupta, and Hrvoje Benko. 2020. Acustico: Surface Tap Detection and Localization using Wrist-based Acoustic TDIOA Sensing. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20)*. Association for Computing Machinery, New York, NY, USA, 406–419. <https://doi.org/10.1145/3379337.3415901>
 - [10] Patrick Grady, Jeremy A Collins, Chengcheng Tang, Christopher D Twigg, Kunal Aneja, James Hays, and Charles C Kemp. 2024. PressureVision++: Estimating Fingertip Pressure from Diverse RGB Images. *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (2024).
 - [11] Patrick Grady, Chengcheng Tang, Samarth Brahmabhatt, 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: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VI*. Springer-Verlag, Berlin, Heidelberg, 328–345. https://doi.org/10.1007/978-3-031-20068-7_19
 - [12] 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 (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 1059–1070. <https://doi.org/10.1145/3332165.3347947>
 - [13] 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 (Dec. 2020), 128:1–128:29. <https://doi.org/10.1145/3432204>
 - [14] 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*. ACM, Santa Barbara California USA, 283–292. <https://doi.org/10.1145/2047196.2047233>
 - [15] 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 (UIST '11)*. Association for Computing Machinery, New York, NY, USA, 441–450. <https://doi.org/10.1145/2047196.2047255>
 - [16] 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>
 - [17] 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 (CHI '10)*. Association for Computing Machinery, New York, NY, USA, 453–462. <https://doi.org/10.1145/1753326.1753394>
 - [18] Zhe He, Xiangyang Wang, Yuanchun Shi, Chi Hsia, Chen Liang, and Chun Yu. 2025. Palmpad: Enabling Real-Time Index-to-Palm Touch Interaction with a Single RGB Camera. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 551, 16 pages. <https://doi.org/10.1145/3706598.3714130>
 - [19] David Heaney. 2024. Quest 3 Has Better Low-Light Hand Tracking Than Quest 3. <https://www.uploadvr.com/quest3s-hand-tracking-better-than-quest-3/>
 - [20] Dan Hendrycks and Kevin Gimpel. 2023. Gaussian Error Linear Units (GELUs). <https://doi.org/10.48550/arXiv.1606.08415> [cs].
 - [21] 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 (Dec. 2020), 132:1–132:24. <https://doi.org/10.1145/3432202>
 - [22] Samuel A. Iacolina, Alessandro Soro, and Riccardo Scateni. 2011. Improving FTIR based multi-touch sensors with IR shadow tracking. In *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems*. ACM, Pisa Italy, 241–246. <https://doi.org/10.1145/1996461.1996529>
 - [23] Daisuke Iwai and Kosuke Sato. 2005. Heat sensation in image creation with thermal vision. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology (ACE '05)*. Association for Computing Machinery, New York, NY, USA, 213–216. <https://doi.org/10.1145/1178477.1178510>
 - [24] Wolf Kienzle and Ken Hinckley. 2014. LightRing: always-available 2D input on any surface. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, Honolulu Hawaii USA, 157–160. <https://doi.org/10.1145/2642918.2647376>
 - [25] Wolf Kienzle, Eric Whitmire, Chris Rittaler, and Hrvoje Benko. 2021. ElectroRing: Subtle Pinch and Touch Detection with a Ring. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3411764.3445094>
 - [26] Daehwa Kim, Keunwoo Park, and Geehyuk Lee. 2021. AtaTouch: Robust Finger Pinch Detection for a VR Controller Using RF Return Loss. 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 11, 9 pages. <https://doi.org/10.1145/3411764.3445442>
 - [27] Daehwa Kim, Eric Whitmire, Roger Boldu, Wolf Kienzle, and Hrvoje Benko. 2024. SoundScroll: Robust Finger Slide Detection Using Friction Sound and Wrist-Worn Microphones. In *Proceedings of the 2024 ACM International Symposium on Wearable Computers (ISWC '24)*. Association for Computing Machinery, New York, NY, USA, 63–70. <https://doi.org/10.1145/3675095.3676614>
 - [28] Myron W. Krueger, Thomas Gionfriddo, and Katrin Hinrichsen. 1985. VIDEO-PLACE—an artificial reality. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '85)*. Association for Computing Machinery, New York, NY, USA, 35–40. <https://doi.org/10.1145/317456.317463>
 - [29] Daniel Kurz. 2014. Thermal touch: Thermography-enabled everywhere touch interfaces for mobile augmented reality applications. In *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 9–16. <https://doi.org/10.1109/ISMAR.2014.6948403>
 - [30] Gierad Laput and Chris Harrison. 2019. SurfaceSight: A New Spin on Touch, User, and Object Sensing for IoT Experiences. 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/10.1145/3290605.3300559>
 - [31] 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 (CHI '11)*. Association for Computing Machinery, New York, NY, USA, 2565–2574. <https://doi.org/10.1145/1978942.1979317>
 - [32] Chen Liang, Xutong Wang, Zisu Li, Chi Hsia, Mingming Fan, Chun Yu, and Yuanchun Shi. 2023. ShadowTouch: Enabling Free-Form Touch-Based Hand-to-Surface Interaction with Wrist-Mounted Illuminant by Shadow Projection. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23)*. Association for Computing Machinery, New York, NY, USA, 1–14. <https://doi.org/10.1145/3586183.3606785>
 - [33] Chen Liang, Chun Yu, Yue Qin, Yuntao Wang, and Yuanchun Shi. 2021. DualRing: Enabling Subtle and Expressive Hand Interaction with Dual IMU Rings. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 3 (Sept. 2021), 1–27. <https://doi.org/10.1145/3478114>
 - [34] Joe Marshall, Tony Pridmore, Mike Pound, Steve Benford, and Boriana Koleva. 2009. Pressing the Flesh: Sensing Multiple Touch and Finger Pressure on Arbitrary Surfaces. In *Proceedings of the 6th International Conference on Pervasive Computing (Pervasive '08)*. Springer-Verlag, Berlin, Heidelberg, 38–55. https://doi.org/10.1007/978-3-540-79576-3_3
 - [35] Damien Masson, Alix Goguet, 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 (UIST '17)*. Association for Computing Machinery, New York, NY, USA, 41–48. <https://doi.org/10.1145/3126594.3126619>
 - [36] Takashi Matsubara, Naoki Mori, Takehiro Niikura, and Shun'ichi Tano. 2017. Touch detection method for non-display surface using multiple shadows of finger. In *2017 IEEE 6th Global Conference on Consumer Electronics (GCCE)*. 1–5. <https://doi.org/10.1109/GCCE.2017.8229364>
 - [37] 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> ISSN: 2642-5254.
 - [38] Vimal Mujibiya, Chris Harrison. 2024. EgoTouch: On-Body Touch Input Using AR/VR Headset Cameras. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology (UIST '24)*. Association for Computing Machinery, New York, NY, USA, 1–11. <https://doi.org/10.1145/3654777.3676455>
 - [39] Adiyan Mujibiya, Xiang Cao, Desney S. Tan, Dan Morris, Shwetak N. Patel, and Jun Rekimoto. 2013. The sound of touch: on-body touch and gesture sensing

- based on transdermal ultrasound propagation. In *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces (ITS '13)*. Association for Computing Machinery, New York, NY, USA, 189–198. <https://doi.org/10.1145/2512349.2512821>
- [40] 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>
- [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. https://doi.org/10.1007/978-3-319-58750-9_26
- [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> Conference Name: IEEE Access.
- [43] Makoto Ono, Buntarou Shizuki, and Jiro Tanaka. 2013. Touch & activate: adding interactivity to existing objects using active acoustic sensing. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, St. Andrews Scotland, United Kingdom, 31–40. <https://doi.org/10.1145/2501988.2501989>
- [44] Pak-Kiu Chung, Bing Fang, and F. Quek. 2008. MirrorTrack - a vision based multi-touch system for glossy display surfaces. In *5th International Conference on Visual Information Engineering (VIE 2008)*. IEE, Xi'an, China, 571–576. <https://doi.org/10.1049/cp:20080379>
- [45] Siyao Pei, Pradyumna Chari, Xue Wang, Xiaoying Yang, Achuta Kadambi, and Yang Zhang. 2022. ForceSight: Non-Contact Force Sensing with Laser Speckle Imaging. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) (UIST '22). Association for Computing Machinery, New York, NY, USA, Article 25, 11 pages. <https://doi.org/10.1145/3526113.3545622>
- [46] DT Pham, M Al-Kutubi, Z Ji, M Yang, Z Wang, and S Catheline. 2005. Tangible acoustic interface approaches. In *Proceedings of IPROMS 2005 Virtual Conference*. Citeseer, 497–502.
- [47] Erez Posner, Nick Starzicki, and Eyal Katz. 2012. A single camera based floating virtual keyboard with improved touch detection. In *2012 IEEE 27th Convention of Electrical and Electronics Engineers in Israel*. 1–5. <https://doi.org/10.1109/EEEI.2012.6377072>
- [48] Mark Richardson, Matt Durasoff, and Robert Wang. 2020. Decoding Surface Touch Typing from Hand-Tracking. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. ACM, Virtual Event USA, 686–696. <https://doi.org/10.1145/3379337.3415816>
- [49] Vivian Shen, James Spann, and Chris Harrison. 2021. FarOut Touch: Extending the Range of ad hoc Touch Sensing with Depth Cameras. In *Symposium on Spatial User Interaction*. ACM, Virtual Event USA, 1–12. <https://doi.org/10.1145/3485279.3485281>
- [50] Xiyuan Shen, Chun Yu, Xutong Wang, Chen Liang, Haozhan Chen, and Yuanchun Shi. 2024. MouseRing: Always-available Touchpad Interaction with IMU Rings. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24)*. Association for Computing Machinery, New York, NY, USA, 1–19. <https://doi.org/10.1145/3613904.3642225>
- [51] Yilei Shi, Haimo Zhang, Jiashuo Cao, and Suranga Nanayakkara. 2020. VersaTouch: A Versatile Plug-and-Play System that Enables Touch Interactions on Everyday Passive Surfaces. In *Proceedings of the Augmented Humans International Conference (AHS '20)*. Association for Computing Machinery, New York, NY, USA, 1–12. <https://doi.org/10.1145/3384657.3384778>
- [52] 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. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 2 (June 2020), 1–25. <https://doi.org/10.1145/3397309>
- [53] Garth Shoemaker, Anthony Tang, and Kellogg S. Booth. 2007. Shadow reaching: a new perspective on interaction for large displays. *Proceedings of the 20th annual ACM symposium on User interface software and technology* (Oct. 2007), 53–56. <https://doi.org/10.1145/1294211.1294221> Conference Name: UIST'07: The 20th Annual ACM Symposium on User Interface Software and Technology ISBN: 9781595936790 Place: Newport Rhode Island USA Publisher: ACM.
- [54] Paul Strelj, Jiaxi Jiang, Juliette Rossie, and Christian Holz. 2023. Structured Light Speckle: Joint Ego-Centric Depth Estimation and Low-Latency Contact Detection via Remote Vibrometry. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. ACM, San Francisco CA USA, 1–12. <https://doi.org/10.1145/3586183.3606749>
- [55] Paul Strelj, Mark Richardson, Fadi Botros, Shugao Ma, Robert Wang, and Christian Holz. 2024. TouchInsight: Uncertainty-aware Rapid Touch and Text Input for Mixed Reality from Egocentric Vision. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. ACM, Pittsburgh PA USA, 1–16. <https://doi.org/10.1145/3654777.3676330>
- [56] Joshua Strickon and Joseph Paradiso. 1998. Tracking hands above large interactive surfaces with a low-cost scanning laser rangefinder. In *CHI '98 Conference Summary on Human Factors in Computing Systems (CHI '98)*. Association for Computing Machinery, New York, NY, USA, 231–232. <https://doi.org/10.1145/286498.286719>
- [57] 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>
- [58] Joseph Thomas. 2013. A Camera Based Virtual Keyboard with Touch Detection by Shadow Analysis. (2013). jsthomass.github.io/docs/vkeyboard/vkeyboard.pdf
- [59] Pavan Kumar Anasosal Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. 2023. FastViT: A Fast Hybrid Vision Transformer using Structural Reparameterization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.
- [60] Boga Vishal and K Deepak Lawrence. 2017. Paper piano – Shadow analysis based touch interaction. In *2017 2nd International Conference on Man and Machine Interacting (MAMI)*. 1–6. <https://doi.org/10.1109/MAMI.2017.8307890>
- [61] Huateng Vision. 2023. HT-SUA33GM-T1V-C : Huateng Vision. <https:////huatengvision.com/product/195/>
- [62] Anandghan Waghmare, Youssef Ben Taleb, Ishan Chatterjee, Arjun Narendra, and Shwetak Patel. 2023. Z-Ring: Single-Point Bio-Impedance Sensing for Gesture, Touch, Object and User Recognition. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. ACM, Hamburg Germany, 1–18. <https://doi.org/10.1145/3544548.3581422>
- [63] Wikipedia. 2025. Lux. <https://en.wikipedia.org/w/index.php?title=Lux&oldid=1284696029> Page Version ID: 1284696029
- [64] 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*. ACM, Seattle WA USA, 83–92. <https://doi.org/10.1145/1095034.1095047>
- [65] Andrew D. Wilson. 2010. Using a depth camera as a touch sensor. In *ACM International Conference on Interactive Tabletops and Surfaces*. ACM, Saarbrücken Germany, 69–72. <https://doi.org/10.1145/1936652.1936665>
- [66] Ziyi Xia, Xincheng Huang, Sidney S Fels, and Robert Xiao. 2025. HaloTouch: Using IR Multi-Path Interference to Support Touch Interactions with General Surfaces. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 548, 17 pages. <https://doi.org/10.1145/3706598.3714179>
- [67] Robert Xiao, Scott Hudson, and Chris Harrison. 2016. DIRECT: Making Touch Tracking on Ordinary Surfaces Practical with Hybrid Depth-Infrared Sensing. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces*. ACM, Niagara Falls Ontario Canada, 85–94. <https://doi.org/10.1145/2992154.2992173>
- [68] Robert Xiao, Scott Hudson, and Chris Harrison. 2017. Supporting Responsive Cohabitation Between Virtual Interfaces and Physical Objects on Everyday Surfaces. *Proc. ACM Hum.-Comput. Interact.* 1, EICS, Article 12 (June 2017), 17 pages. <https://doi.org/10.1145/3095814>
- [69] Robert Xiao, Julia Schwarz, Nick Throm, Andrew D. Wilson, and Hrvoje Benko. 2018. MRTouch: Adding Touch Input to Head-Mounted Mixed Reality. *IEEE Transactions on Visualization and Computer Graphics* 24, 4 (April 2018), 1653–1660. <https://doi.org/10.1109/TVCG.2018.2794222>
- [70] 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 (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 5126–5137. <https://doi.org/10.1145/2858036.2858286>
- [71] Cheng Zhang, Qiuyue Xue, Anandghan Waghmare, Sumeet Jain, Yiming Pu, Sinan Hersek, Kent Lyons, Kenneth A. Cunefare, Omer T. Inan, and Gregory D. Abowd. 2017. SoundTrak: Continuous 3D Tracking of a Finger Using Active Acoustics. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 2 (June 2017), 1–25. <https://doi.org/10.1145/3090095>
- [72] 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> arXiv:2006.10214 [cs].
- [73] Lymn Zhang, Anyi Rao, and Maneesh Agrawala. 2025. Scaling In-the-Wild Training for Diffusion-based Illumination Harmonization and Editing by Imposing Consistent Light Transport. In *The Thirteenth International Conference on Learning Representations*. <https://openreview.net/forum?id=u1cQYxR1H>
- [74] 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 (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 1151–1159. <https://doi.org/10.1145/3332165.3347869>
- [75] Yang Zhang, Junhan Zhou, Gerard Laput, and Chris Harrison. 2016. SkinTrack: Using the Body as an Electrical Waveguide for Continuous Finger Tracking on the Skin. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing*

- Systems*. ACM, San Jose California USA, 1491–1503. <https://doi.org/10.1145/2858036.2858082>
- [76] Yiming Zhao, Taein Kwon, Paul Strelci, Marc Pollefeys, and Christian Holz. 2025. EgoPressure: A Dataset for Hand Pressure and Pose Estimation in Egocentric Vision. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*. 27727–27738.