



HaloTouch: Using IR Multi-Path Interference to Support Touch Interactions with General Surfaces

Ziyi Xia

Department of Computer Science
University of British Columbia
Vancouver, British Columbia, Canada
zxia0101@cs.ubc.ca

Sidney S Fels

Electrical and Computer Engineering
University of British Columbia
Vancouver, British Columbia, Canada
ssfels@ece.ubc.ca

Xincheng Huang

Department of Computer Science
University of British Columbia
Vancouver, British Columbia, Canada
xincheng.huang@ubc.ca

Robert Xiao

Department of Computer Science
University of British Columbia
Vancouver, British Columbia, Canada
brx@cs.ubc.ca

Abstract

Sensing touch on arbitrary surfaces has long been a goal of ubiquitous computing, but often requires instrumenting the surface. Depth camera-based systems have emerged as a promising solution for minimizing instrumentation, but at the cost of high touch-down detection error rates, high touch latency, and high minimum hover distance, limiting them to basic tasks. We developed HaloTouch, a vision-based system which exploits a multipath interference effect from an off-the-shelf time-of-flight depth camera to enable fast, accurate touch interactions on general surfaces. HaloTouch achieves a 99.2% touch-down detection accuracy across various materials, with a motion-to-photon latency of 150 ms. With a brief (20s) user-specific calibration, HaloTouch supports millimeter-accurate hover sensing as well as continuous pressure sensing. We conducted a user study with 12 participants, including a typing task demonstrating text input at 26.3 AWPM. HaloTouch shows promise for more robust, dynamic touch interactions without instrumenting surfaces or adding hardware to users.

CCS Concepts

- Human-centered computing → Interaction techniques.

Keywords

Input Techniques, Touch Sensing, Ubiquitous Computing, Pressure and Proximity Sensing, Ad-hoc Surface Interaction, On-world Interaction

ACM Reference Format:

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 *CHI Conference on Human Factors in Computing Systems (CHI*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CHI '25, Yokohama, Japan

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-1394-1/25/04
<https://doi.org/10.1145/3706598.3714179>

'25), April 26–May 01, 2025, Yokohama, Japan. ACM, New York, NY, USA,
17 pages. <https://doi.org/10.1145/3706598.3714179>

1 Introduction

As computing becomes increasingly ubiquitous, it is moving beyond the confines of small screens and dedicated devices. This evolution is paving the way for “on-world” interfaces that leverage the physical environment itself for interaction [31, 62]. Traditional input methods, such as mice and keyboards, though precise, are tied to specific devices and lack the flexibility required for spontaneous, ad-hoc use in dynamic settings. Recent advances in natural input techniques, like hand gestures [14, 29, 60] and speech recognition [4, 35, 55], have enabled more intuitive and controller-free interactions. However, these methods are often limited by a lack of haptic feedback [40, 47], reduced precision [11, 46], and user fatigue [6, 28] over extended periods. To overcome these challenges, the focus is now on developing robust input sensing techniques that can transform everyday surfaces into interactive, touch-sensitive interfaces.

Several techniques have been proposed to enable touch input on various surfaces. For instance, Electrick [69] uses electric field tomography to achieve remarkably high touch accuracy. Taplight [58] reduces touch latency to an unprecedented 50.4ms by analyzing structured light patterns. ShadowTouch [39] has a *touch point threshold* of less than 2mm – accurately triggering touch down events only when the finger is within 2mm of the surface. Tri-pad [16] takes a human-centric approach to generalize touch input across a wide range of surface materials. Meanwhile, Micropress [15] combines pressure and proximity sensing by integrating an IMU (Inertial Measurement Unit) into the fingers. While each of these innovative techniques excels in one or two specific metrics, none of them can comprehensively address the entire design space of touch input (see Figure 1).

We present **HaloTouch**, a novel technique enabling comprehensive sensing input capabilities with a commercially available depth camera. **HaloTouch** achieves a touch-down accuracy of 99.2% across 5 different materials, spatial accuracy within 5.5mm, a touch point threshold of 4.97mm, and a motion-to-photon latency

of 150ms. Furthermore, the pressure and proximity detection capabilities showed high reliability, with mean pressure error of 18.77% and mean proximity error of 2.81 mm.

HaloTouch leverages the underexplored phenomenon of depth camera multipath interference, enabling it to work on general day-to-day surfaces. Multipath interference provides a continuous signal around the fingertip area as the finger approaches the surface, enabling us to determine the fingertip's proximity to the surface despite sensor noise. We refer to this phenomenon as the Halo effect, as it visually appears as a “halo” around the fingertip. This technique requires no surface instrumentation or wearable sensors, though we do need to perform a short (20s) one-time calibration for users on a new material. The unique Halo phenomenon also enables touch input in challenging conditions, such as very wet surfaces.

After comparing with other related work, we came up with design requirements for **HaloTouch**. We describe the theory behind multipath interference, the Halo effect and our approach to model this signal for building a touch detection pipeline. We then report on a two part study, the first one capturing the technical evaluations of **HaloTouch** and the second one showing the interaction modalities it enables with passive surfaces.

Finally, we summarize our contributions as follows:

- Explored and characterized the Halo effect as a way to surpass noise thresholds in commodity depth sensors.
- Developed a comprehensive multimodal touch input system, supporting both pressure and proximity sensing, for use on passive physical surfaces using only a depth+RGB camera.
- Demonstrated the versatility and effectiveness of our system for rapid, ad hoc use cases through a user study focused on typing, alongside two additional example applications.

2 Related Work

Prior approaches, including surface instrumentation, wearables, and vision-based methods, offer trade-offs in accuracy, scalability, and practicality. While depth-sensing and computer vision provide promising alternatives, existing methods often struggle with real-time performance or deployment complexity. This section reviews prior work in ad-hoc touch, pressure, and proximity sensing.

2.1 Enabling Ad-hoc Touch Input on Passive Physical Surfaces

Researchers have explored various methods for capturing touch events on everyday surfaces. One approach is surface instrumentation with acoustic sensors or electrodes. For example, prior research has localized touch events on surfaces with signals from contact microphones [26, 34, 51, 65]. Similarly, electrodes can also be added to objects or surfaces to make them touch-sensitive [7, 69, 70]. However, instrumenting surfaces introduces deployment overhead and is costly when scaling to additional surfaces. Other research has explored instrumenting users with wearables to detect touch signals. SkinTrack [71] integrates four electrodes into a smartwatch, localizing finger touches by measuring phase differences of active electrical signals emitted from a separate finger-worn device. ShadowTouch [39] adds a light emitter to a user's wrist and predicts subtle touch events by analyzing the shadow of the user's hand.

Additionally, prior research has developed small finger-mounted IMUs to detect touch actions during interactions. Gu et al. [23] used an IMU in a finger ring to achieve low latency touch detection, whereas Oh et al. [50] mounted the IMU directly on the fingertip for high accuracy touch contact detection. ActualTouch [57] mounted a small IMU on the user's fingertip to detect differences in finger microvibrations between touching/dragging and in-air states.

Camera-based approaches for sensing ad-hoc touch have become popular as they do not require instrumenting the surface nor the user. Wilson [61] was the first to demonstrate the possibility of using a depth camera to detect touches on flat surfaces. Other explorations have focused on improving different metrics of touch input, such as providing high touch down accuracy [25, 64], low spatial error [8, 27, 37], low latency [17, 58], and longer sensing range [56]. Among those, DIRECT [64] combined depth and IR signal sources to reach more than 99% touch down accuracy at the cost of higher latency. TapLight [58] leveraged structured laser light reflection disparity to reach a very low latency (50.4 ms) but their touch point threshold is as high as 45 mm.

Deploying ad-hoc touch input systems on headsets provides precise, tactile input for augmented and virtual reality applications [30, 32], complementing existing natural input techniques. MRTouch [66] achieved real-time surface plane detection and touch input detection using sensors embedded in HoloLens 2. It demonstrated the potential for fast touch input interactions using IR and depth-sensing techniques, although no formal evaluation was conducted to assess how rapid users can give touch inputs to the system. Tripad [16] used a more human-centric approach to generalize touch input to arbitrary surfaces using only hand tracking from HoloLens 2. It successfully performed basic input tasks like clicking and dragging, though the touch point threshold remains relatively high at 55 mm, making it difficult to rapidly disengage touches for fast input. EgoTouch [49] uses Apple Vision Pro's RGB cameras for calibration-free, on-skin touch detection, achieving high accuracy in various environments, but it struggled with high-speed input and occlusions. TouchInsight [59] employed a probabilistic framework to handle egocentric hand tracking uncertainties, enabling precise, low-latency interactions like virtual keyboard typing. In contrast, **HaloTouch** combines pressure and proximity sensing with accurate touch input across general surfaces, expanding interaction capabilities.

2.2 Ad hoc Pressure Sensing and Proximity Sensing

Traditional pressure sensing methods involve physical sensors such as capacitive grids [2, 12, 24], force-sensitive resistors [5, 53], or flexible sensors [3, 36], which must be mounted on surfaces or hands. These methods can be impractical and expensive for diverse, real-world applications. Vision-based surface pressure sensing is a growing field that aims to estimate contact pressure using computer vision techniques rather than invasive physical sensors.

Recent advances in computer vision have allowed for pressure estimation using visual cues such as fingertip color changes [9, 43, 44]



Figure 1: Radar charts comparing eight touch input systems (Electrick[69], ShadowTouch[39], Shi et al.[57], TapLight[58], Direct[64], FarOut[56], MicroPress[15], HaloTouch) across metrics like Accuracy, Latency, Material Compatibility

and soft tissue deformation [33]. ForceSight [52] showed high pressure sensing potentials using laser speckle pattern differences reflected from deformed objects, though it only works with plastically deformable materials. PressureVision [22] pioneered using deep learning to estimate fingertip pressure from RGB images, but it was limited by controlled settings with constrained lighting and simple surfaces. To address these limitations, the “PressureVision++” [21] model utilized weak supervision through contact labels. This method requires the collection of diverse training data without requiring ground-truth pressure measurements from high-resolution sensors. In comparison, **HaloTouch** leverages learning of the IR and depth multipath interference signal, avoiding the use of large training set to estimate a more fine-grained pressure level on flat surfaces.

Pressure sensing and proximity sensing can be achieved at the same time to enable new modes of interaction. MicroPress [15] used Inertial Measurement Unit (IMU) sensors combined with deep learning models to detect both pressure and hover distance between fingers with high precision. However, this method requires additional sensors to be placed on both fingers, potentially limiting its practicality for casual or on-the-go use. Other existing approaches focus on finger proximity detection in mobile interactions. Air+Touch [10] employed a horizontally mounted depth camera on a smartphone to explore the interaction space for both touch and hover gestures. Similarly, OmniSense [67] used a 360-degree camera feed on a mobile phone to train a neural network that predicts the 3D location of fingertips. Another technique proposed by Matulic [45] involved two mirrors mounted above the phone screen that reflect the front camera view; captured video is sent to a

deep neural network that robustly infers the 3D position of fingertips. Although effective, these methods still require some level of instrumentation, either on the touch surfaces or the mobile devices, which makes achieving spontaneous, ad hoc proximity sensing difficult. **HaloTouch** uses only a commodity available depth camera, offering the potential to achieve ad hoc proximity sensing and pressure sensing for any surfaces anywhere.

3 Positioning and Goals of HaloTouch

We identified representative touch sensing technologies with high reported accuracy (above 95%) and distilled eight key technical dimensions. From these, we selected works that demonstrated the best performance in each dimension to ensure a comparison against state-of-the-art benchmarks. Finally, we evaluated and mapped the overall performance of the selected works across these eight dimensions. The dimensions and their scoring criteria are detailed below, with additional scoring information provided in Appendix Figure 15.

- **Instrumentation:** Past touch sensing systems are mainly based on 3 types of instrumentation: 1) instrumenting surfaces, 2) wearables, and 3) vision-based (i.e., use headsets or Kinect). To score them, we consider deployment friction, scalability, and interactive freedom. Systems that require instrumenting the surface have a lower score because they require the most effort to deploy and are hard to scale. We assign technologies that instrument users’ hands with wearables a medium score as they can be bulky and thus affect interactive freedom. Consequently, we consider vision-based technology to have lower deployment friction and better interactive freedom.

- *Touch Down Accuracy*: We extract the touch down accuracy reported from the corresponding literature in percentages and normalize them to 0–10 (0% has a score of 0 and 100% has a score of 10).
- *Spatial Accuracy*: We extract the spatial accuracy from the corresponding literature in mean errors of millimeters, and min–max normalize them to 0–10. Some work (e.g., ShadowTouch [39], Gu et al. [23], Shi et al. [57], and TapLight [58]) only focus on touch down accuracy. Such work can be combined with built-in hand tracking for touch localization; we thus assign a typical hand tracking spatial accuracy of 11 mm [1] to them for scoring.
- *Latency*: We extract the latency from the corresponding literature in milliseconds, and min–max normalize them to 0–10.
- *Touch Point Threshold*: We define the touch point threshold as the minimum distance the fingertip must travel perpendicular to the surface to register a valid input, while avoiding false positives. We extract the touch point threshold from the corresponding literature in millimeters, and min–max normalize them to 0–10. For work that did not report a specific touch point threshold (i.e., Electrick [69], Gu et al. [23], Shi et al. [57]), we estimated it from their technical principle and demo video.
- *Material Compatibility*: This is an under-explored yet important dimension. Our score is based on our interpretation of their technical principle. For example, technologies that require conductive material [69] or plain texture of surfaces [39] have a lower material compatibility.
- *Pressure*: While novel track pads [54] already support pressure sensing for touch, enabling the same on everyday surfaces is scarce in literature.
- *Proximity*: For our evaluation, we selected a recent high-performance system [15] as a benchmark and assessed our system based on their proximity accuracy metrics.

We compare past touch-sensing research with their performance in common metrics distilled from literature. We select representative research that falls in the instrumentation categories we have reviewed: 1) instruments the surface with sensors, 2) instruments the user with wearables, and 3) instrumentation free (besides using RGB/Depth sensing equipment). State-of-the-art touch sensing has reached 99% accuracy and millimeter-level spatial accuracy while having a low latency. However, few (only ShadowTouch [39] to the best of our knowledge) have focused on achieving a small touch point threshold, enabling subtle touch detection. With **HaloTouch**, we set out to minimize the touch point threshold while achieving similar touch-event and spatial accuracy to the state-of-the-art, enabling ad hoc, rapid touch interactions on passive physical surfaces. As such, we list the technical requirements for our system:

- (1) Avoid instrumenting users or the surface with additional hardware, thus supporting better interaction freedom and low-friction deployment.
- (2) Achieve high accuracy in detecting touch-down moments to ensure that every intentional touch by the user is correctly identified and registered by the system.

- (3) Provide high spatial accuracy in touch detection, allowing the system to precisely identify the exact location of each touch input.
- (4) Ensure the interaction system responds to touch inputs with a low latency to maintain a fluid and responsive user experience.
- (5) Minimize the touch point threshold to allow for closely spaced touch inputs without interference. This feature is particularly important in scenarios where rapid or complex gestures are performed, such as fast tapping on multiple locations on the surface.
- (6) Support operation across diverse surface materials to enhance usability in various mobile environments.

HaloTouch builds upon the under-explored phenomenon of multipath interference in depth cameras, first highlighted by FarOut [56]. While FarOut utilized this phenomenon along with additional engineering techniques to extend touch detection range up to 3 meters, our work provides a more detailed explanation of the underlying principles and a comprehensive characterization of multipath interference for touch sensing with depth cameras. Furthermore, our system extends the capabilities of touch input beyond mere contact detection to include hover and pressure inputs, leveraging multipath interference to enable these additional interaction dimensions.

4 HaloTouch

In this section, we first explain the relationship between multipath interference and the Halo effect, and the verification for our theory. Then we detail how we implement our systems to take advantage of the Halo effect to achieve touch sensing.

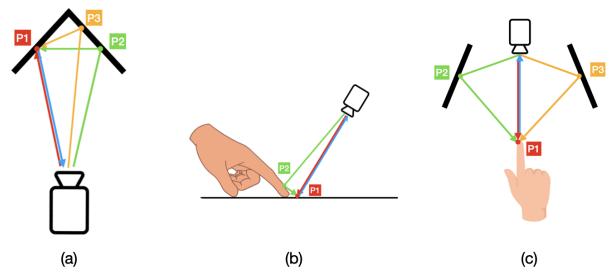


Figure 2: Multi-path Interference example scene: (a) Top view of Multi-path interference when camera faces a corner. (b) Side view of multi-path interference when camera faces a hand. (c) Top view of multi-path interference when camera faces a hand.

4.1 Working Principal

4.1.1 Multipath Interference in Time-of-flight Depth Camera. Modern time-of-flight depth cameras rely on correlation techniques to determine the phase shift, and thereby the distance between two signals: a modulated infrared signal that illuminates

the scene and the signal received by the sensor, which captures the reflection from the scene's geometry [38]. Unlike other depth sensing techniques, the time-of-flight depth camera contains an active IR light source that emits a cone of IR light and receives the reflected IR signal using a sensor array to produce a depth map of a 3D scene.

Due to the diffuse reflection pattern of most objects in our life, the same incident light ends up creating multiple light paths while hitting objects with complex geometries. This causes several signals with different path lengths to converge on the same pixel, interfering with each other. Since the sum of two phase-shifted sine waves is itself a phase-shifted sine wave, this interference leads the camera to register incorrect depth measurements, potentially resulting in significant errors in both depth and amplitude – the multipath interference [18].

We explain the multipath interference phenomena in Figure 2(a), when a time-of-flight depth camera illuminates an IR light (red line) into a corner. After the IR signal hits the first contact point (P1), it reflects in many directions following a combination of specular reflection (not drawn) and diffuse reflection (blue line) [13]. At the same time, the other incident lights hit point P3, creating specular reflection (green line) and hit point P2, creating diffuse reflection (yellow line). Their reflection paths all converge back to the same sensor array cell corresponding to P1, resulting in different phase shifted signals, and thus different computed distances after each signal is correlated with the reference signal. We note that there are typically multiple diffuse reflection sources that converge to P1 in practice, and they all contribute to the final signal strength observed at P1. Therefore, the multipath interference phenomenon generates a complex mixed signal influenced by multiple factors. Additionally, for smooth surfaces that only create specular reflections, the multipath interference phenomena will not be observed since each contact point (P1) will only receive one unique specular reflection coming from the reflection point (P3).

4.1.2 Multipath Interference in Touch Down Events. Next, we explain the multipath interference phenomenon in the context of a finger touch-down event, influenced by both local and global impacts. First, when a finger approaches the surface closely (within 3 cm), the finger and the target surface form a mini corner, creating local interference (Figure 2(b)). At this corner, multipath interference occurs in the same manner as previously described: the red line represents the incident light, which results in diffuse reflections along the blue line after hitting point P1. The other incident light ray (green line) hits point P2 and converges with the blue line path to form multipath interference (reflection angle exaggerated in the figure).

In addition to this local impact, we illustrate the global impact from a top-down perspective of the same finger touch-down event (Figure 2(c)). Beyond the single incident light ray (red line) directed at point P1, the infrared light source also emits other rays in a cone shape. We consider two rays (green line and yellow line) at the widest angles of this cone, which are directed at points P2 and P3—representing two points on other objects in the scene. In this scenario, diffuse reflections are the most common, as the objects may not have perfectly angled surfaces to produce specular reflections. The reflected rays eventually also reach P1, causing P1

to receive phase-shifted light originating from other objects in the scene. Both local and global influences contribute to the final signal strength observed.

4.1.3 The Halo Effect. The direct impact of multipath interference in many cases takes the form of unbiased noise, which is hard to eliminate. This means that even if we have a higher resolution depth camera, we are limited in our effective depth resolution. Previous researchers had used different techniques to decouple each phase shifted signal to recover the true phase shift; however they have been limited by bandwidth and SNR [18]. Instead of modeling the formation of this phenomena, we leverage a case of biased multipath interference as observed directly in the output signal in the depth map, after the interference is processed by the camera sensor.

The Halo effect is influenced by several factors, including the material of the reflecting surfaces, the angle between the ground surface and the camera, the geometry of the objects, their orientation relative to the camera, and the distance between them, among others. We note that the Halo effect signal is highly sensitive; even slight deformations of the fingertip on the surface can cause noticeable variations in the Halo signal, whereas the changes in the depth data at the fingertip itself are imperceptible due to noise. Our pressure sensing technique leverages this high level of sensitivity. To model the Halo effect, we considered the index finger's six degrees of freedom, incorporating the effects of material properties and finger deformation when pressure is applied to the finger press. We empirically characterized the Halo effect using a single index finger interacting with a surface while isolating each of the parameters mentioned above. Each parameter exhibited a non-linear relationship with the Halo effect signal strength, as expected, since changes in one parameter often lead to multiple concurrent effects. For instance, altering only the finger's x-position while keeping other parameters constant affects both the distance and orientation between the finger and the camera. Given this non-linear relationship, we adopt a machine learning approach to predict the Halo effect signal strength, as described later in Section 4.3.1.

The key observation that supports our multipath interference hypothesis is that, after extracting the output signal from the depth camera, visualizing it on the depth map, and subtracting the background depth, the resulting values are positive. This indicates that the depth camera interprets these points as being further away than the surface actually is; in other words, the camera perceives a “denting” of the surface. Furthermore, this value increases as more interference is introduced into the scene, which aligns with our hypothesis that a mixture of longer-path, phase-shifted signals causes the camera to interpret the pixel as being further away than its actual distance. More technical details are provided in Section 4.3.1.

We considered other potential explanations for this phenomenon, such as signal averaging and over-exposure artifacts. Signal averaging happens when objects are very close to a surface, causing the boundary between them to blur or disappear, but we ruled this out because the phenomenon we observed occurs after signal averaging in the sequence of events. As the fingertip approaches the surface, an additional output pixel signal appears around it when it gets within ~2 cm, following an initial signal loss at ~3

cm due to averaging. Over-exposure artifacts, which distort the depth map on overexposed infrared images, were also considered. However, unlike our observed phenomenon, which only occurs when an object is near the surface, over-exposure artifacts affect a larger area of the scene regardless of object proximity.

4.2 Hardware

HaloTouch is implemented using Microsoft Kinect Azure depth camera [48], a Optoma LDMLTUZST LED projector, and a commercial grade laptop with Intel i7 11800H CPU at 2.3 GHz, and NVIDIA GeForce RTX 3050 Ti GPU. The depth camera is USB connected to the PC and the projector is connected to the PC via HDMI. Each hardware component in our setup is replaceable by other similar off-the-shelf products.

Although there are many types of depth camera sensors, our approach is exclusive to the time-of-flight sensing modality. Such sensors are commonplace, e.g. in the Microsoft Kinect 2, Azure Kinect, and Microsoft Hololens series of devices, and the Halo effect has been observed in each of these devices.

4.3 Software

HaloTouch software is implemented using Python in a Windows Anaconda environment. **HaloTouch** uses the pyk4a python wrapper for Azure Kinect SDK to access the depth and RGB streams of the camera. We use the unbinned Narrow Field of View mode to obtain a 30 FPS, 640 x 576 depth stream and a longer operating range (0.5 – 3.86m), with the trade-off of a relatively narrower field of view (75 degrees x 65 degrees). The RGB streams also runs at 30 FPS, and are dynamically warped to match the resolution of the depth stream.

We visualize our system pipeline in Figure 3, and we provide more details in each section below.

4.3.1 Revealing the Halo Effect. Like other static setup systems, such as WorldKit [63], we capture 60 frames of background depth images to model the static environment. We then subtract incoming depth frames from this background model to isolate moving hands from static objects in the scene. In contrast, DIRECT [64] employs a five-second rolling window of depth data to dynamically update the background model, which is particularly useful for accurately modeling backgrounds when moving objects are present. However, in our case with a largely static background, a simple background capture is sufficient to achieve the desired isolation with lower complexity. After background subtraction, the moving hand or any new object will have non-zero pixel values.

To visualize the Halo effect, we apply a filter that sets all pixel values greater than a threshold value - in our current system, 8 units - to 255 and all the pixel values smaller than 0 to 127. After background subtraction, all objects in the scene show negative pixel values since they are closer to the camera compared to the background surface. The Halo effect, on the other hand, manifests with positive pixel values, indicating that the sensor interprets these pixels as being further away from the camera relative to the background surface. The filtered image renders hands in color grey and Halo effect in color white.

The Halo threshold value of 8 was empirically determined by testing on pilot users and we used this threshold value throughout

our experiments. We recommend using a threshold value between 5-10 depending on the camera placement and surface materials (Figure 16). A much higher threshold value would diminish the captured Halo effect signal strength, while a lower value is more susceptible to ambient noise.

4.3.2 Hand Tracking. Vision-based methods for real-time hand tracking have become increasingly mature, though they still face challenges in certain situations, such as occlusions and variations in lighting conditions. When virtual content is projected onto surfaces, interacting with this content can cause color overlays on the hands in the captured RGB stream images. Our hand tracking relies on Google’s MediaPipe Hand Landmarker [20], a deep learning-based algorithm that can fail abruptly due to color discrepancies on the user’s hands. While it is possible to train a custom hand-tracking model with a dataset featuring overlaid hands, such a model may not generalize well to a broader range of colors when the projected objects change. To address this, we use a computer vision approach that combines infrared (IR) and RGB image streams in a 0.7:0.3 ratio for image blending. This blended image enhances the color contrast between the hands and the projected virtual content, thereby restoring effective hand-tracking functionality.

4.3.3 Halo Effect Signal Extraction. As we warp the color image to be the same position and size as the depth image, we can use hand tracking in the color image to find the fingertip in the depth image. We extract a 30x20 patch around the detected fingertip to perform touch detection. The patch area size is empirically determined to capture the full Halo effect signal around the fingertip, although it may preclude detecting closely-spaced fingertips (a limitation of our implementation). We sum all the non-zero values in this patch to form our initial Halo effect signal value and pass this value to the signal corrector.

4.3.4 Signal Correction. In the context of a touch-down event, many factors could contribute to the Halo effect signal strength in a non-linear manner. Limiting the setup to an empty scene, the Halo effect is influenced by the following factors: the distance between the fingertip and the camera, the 3D orientation of the finger (including roll, pitch, and yaw) relative to the camera, and the amount of finger deformation on the surface. To model these factors, we define four key parameters: x , y , θ_p and θ_y . Our model is based on two key assumptions: (a) users maintain a relatively consistent roll angle θ_r while interacting with the surface, and (b) the distance between the camera and the interaction surface remains constant during touch. The parameters x and y represent the fingertip position in the horizontal plane. The pitch angle θ_p and yaw angle θ_y are derived from the spatial relationship between the fingertip and the fingertip (distal interphalangeal joint - the first joint closest to the fingertip) using hand landmarks. Note that the system’s robustness is constrained to the 15 to 75 degrees range. Angles below 15 degrees result in a weak Halo effect signal that compromises detection reliability, while angles above 75 degrees can lead to hand-tracking issues caused by fingertip occlusion from the top view.

To minimize user calibration time and reduce the need for extensive datasets, we employ a HistGradientBoosting regressor to predict the strength of the Halo effect and correct the original Halo

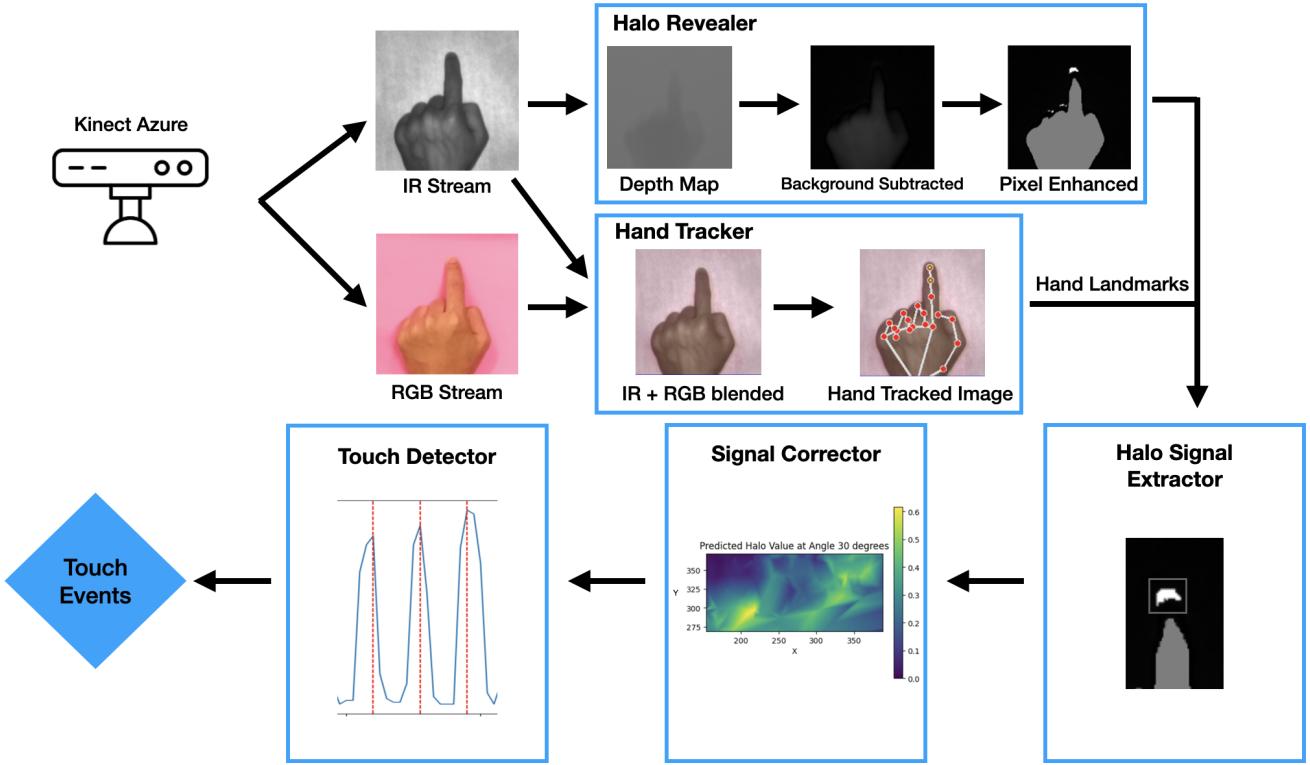


Figure 3: Software: a touch detection pipeline using a Kinect Azure depth camera. It integrates IR and RGB streams with hand tracking and signal processing to isolate fingertip signals, correct noise, and detect touch events

effect signal value, taking x , y , θ_p and θ_y as input, and producing a scalar value as output. The regressor model is initially trained using data collected from three pilot users and is fine-tuned for each new user with 20 seconds of calibration data. We use MSE loss, learning rate of 0.1, boosting iterations of 200, max leaf nodes of 31, min samples leaf of 25, and max bins of 255 for regressor training. Our training data consists of 27000 frames divided into a training-to-validation ratio of 8:2. Three pilot users completed three rounds of sweeping at pitch angles of 15°, 30°, 45°, 60°, and 75° across the surface, as described in Section 4.3.5, Task Two. Each round lasted one minute. During each round, the pilot users were instructed to naturally drag their finger across the surface, accompanied by the 3D-printed blocks, to introduce variations to the yaw angle.

4.3.5 Calibration. The calibration process consists of two main tasks that users must perform to ensure accurate touch detection.

In the first task, users are required to perform four distinct touch states on the surface to calibrate the system for different levels of interaction:

- (1) Hover State: The finger hovers at a height where the Halo effect just becomes visible. This state captures the initial onset of the Halo effect as the finger gets closer to the surface.
- (2) Subtle State: The finger hovers closer to the surface but does not make contact. This state represents a finer level of proximity where the Halo effect becomes more pronounced but still without a physical touch.

- (3) Touch State: The finger lightly touches the surface with a relaxed hand posture. This state is the baseline for a standard touch interaction, where the finger makes contact with the surface without exerting additional pressure.
- (4) Pressure State: The finger presses down on the surface with a medium amount of force. This state represents a more forceful interaction, which helps in distinguishing between a simple touch and a press action.

In the second task, users quickly sweep their fingers around the touch surface. This sweeping motion allows the system to capture the Halo effect strength at various locations across the surface. To maintain consistency in finger deformation while sweeping, a 3D-printed part angled at 45 degrees is provided for users to rest their index finger on. We employed a fixed 45-degree angle to provide consistency in the collected data. However, in principle, any angle from 15 to 75 degrees could be used, as we trained the signal correction model itself on this range of finger angles.

It is important to note that both calibration tasks are only performed using the dominant hand's index finger. We assume that participants have similar hand structures on both sides; therefore, the calibrated parameters are applied to both hands. However, if significant differences are present, as in the case of a participant with substantially longer artificial nails on the dominant hand compared to the non-dominant hand, separate calibration parameters are set for each hand.

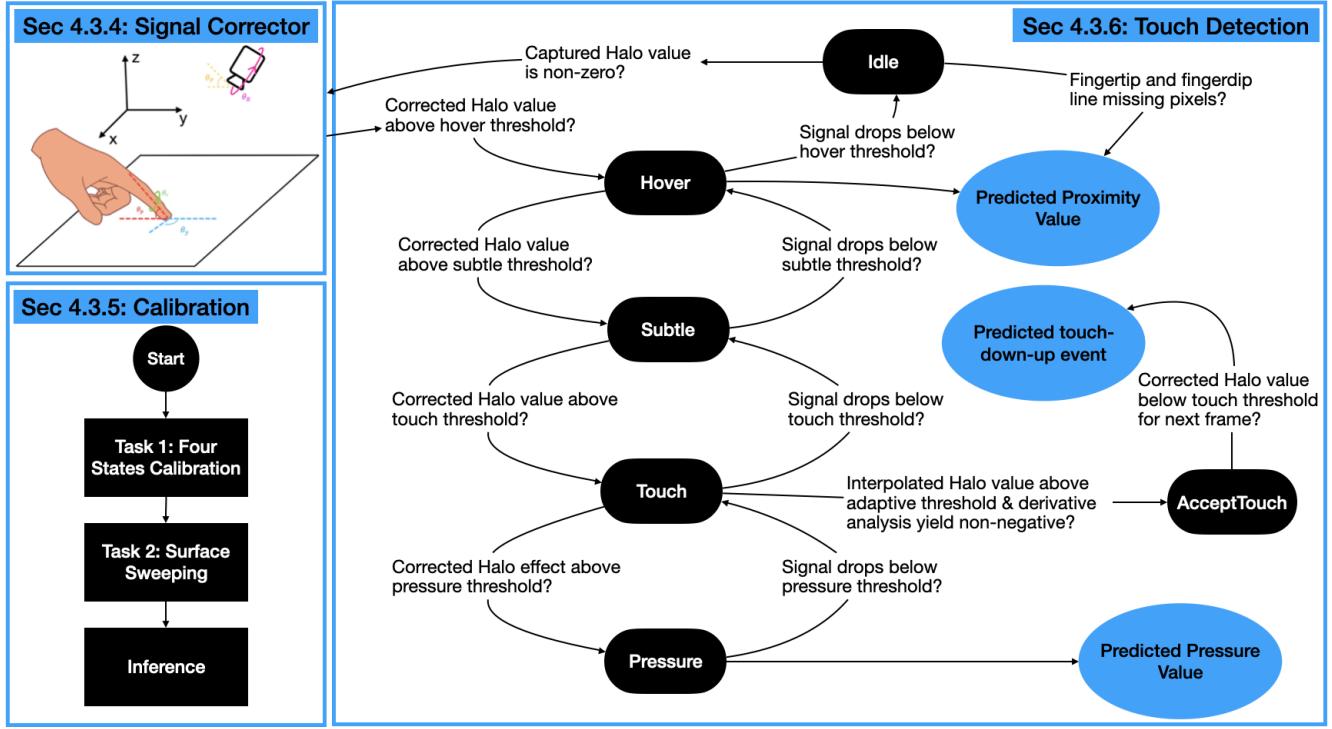


Figure 4: HaloTouch State Machine for Signal Correction, Calibration, and Touch Detection. This diagram outlines the signal processing pipeline for HaloTouch, structured into three key sections: Signal Correction (Sec 4.3.4), Calibration (Sec 4.3.5), and Touch Detection (Sec 4.3.6). The state machine transitions between Idle, Hover, Subtle, Touch, and Pressure states based on corrected halo signal values. The Touch Detection module predicts proximity, touch-down, and pressure values

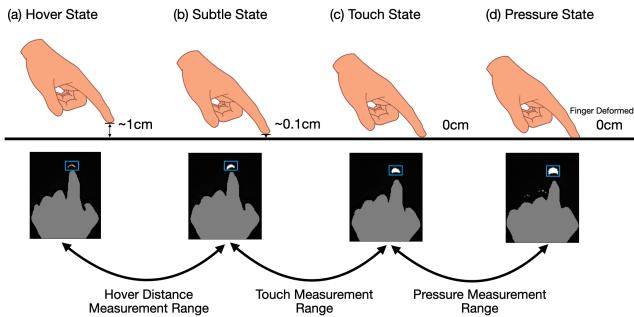


Figure 5: Four calibration states. First row: typical finger to surface distances for each state. Second row: halo effect visualization for each state.

4.3.6 Touch Detection. The touch detector identifies touch, proximity, and pressure using the calibrated values from the four calibration states and follows the same interpolation approach described in Equation 1, where x_{min} and x_{max} represent the calibrated state values, and y_{max} and y_{min} represent the ground truth maximum and minimum for the corresponding measurement.

$$y_{measurement} = y_{min} + \frac{x - x_{min}}{x_{max} - x_{min}} (y_{max} - y_{min}) \quad (1)$$

To detect touch events, the detector interpolates the input Halo effect value x between the subtle state (x_{min}) and the touch state (x_{max}) values to generate an interpolated value. This value is analyzed using a heuristic-based classifier that relies on thresholding and derivative analysis. The classifier calculates the absolute difference between the current interpolated value and an adaptive baseline. If this difference exceeds a predefined threshold, it proceeds to compute the derivative using the difference between the current and previous interpolated values. A non-negative and relatively large magnitude derivative value confirms a potential touch event by identifying an increasing trend, which is the characteristic of a touch down rather than touch up interaction. The baseline is adaptively updated using exponential smoothing to accommodate signal drift and reduce noise. This approach enables accurate touch detection by leveraging sharp deviations and directional changes to minimize noise and false positives. While the derivative analysis requires two consecutive frames to compute changes accurately, the resulting one-frame delay is a small trade-off for the significant improvement in detection accuracy and reliability.

For typing, a logical state machine debounces touch events, only classifying each touch down and up sequence as a single key input.

It verifies valid sequences by requiring touch detection to end with another no-touch detection, reducing false positives during rapid interactions. Although it adds another frame of delay, this approach ensures robust detection, avoiding issues with dropped frames and systematic delays.

The system predicts hover distance by interpolating between the hover and subtle state values using Equation 1. Distances just beyond the hover state (10–15 mm) are predicted by interpolating how many pixels still remain on the line between the fingertip and the fingerdip. Similarly, pressure is predicted by interpolating between touch and pressure state values, enabling differentiation between light and firm touch interactions.

4.4 Camera Placement Impact on Signal Strength

Camera placement affects the strength of the Halo effect signal because changes in yaw angle, pitch angle, and vertical (z) distance alter the intensity and angle of the IR illumination on the surface. We exclude roll angle from consideration because the camera emits a symmetrically distributed IR cone, making roll variations insignificant for signal strength. Figure 6 illustrates how yaw angle influences the Halo effect signal strength based on three trials collected at each incremental camera position with the finger placed at the center of the frame. The impact of pitch angle and vertical (z) distance is presented in Appendix B.

It is important to note that system performance depends on the relative differences between calibration states rather than on absolute signal strength. Additional discussion of signal strength and its implications for sensing flexibility can be found in Section 7.4.

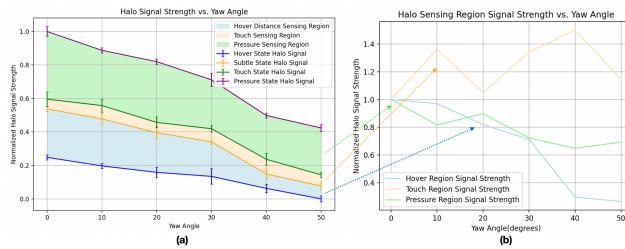


Figure 6: Impact of Camera Yaw Angle on Signal Strength: (a) The Halo signal strength exhibits a similar pattern across all four calibration states when the camera yaw angle ranges from 0 to 50 degrees. (b) Sensing region signal strength: Each sensing region height was calculated and normalized to its 0 degree value

4.5 Working under Extreme Conditions

The Halo effect also provides reliable signals in challenging conditions, such as on watery surfaces. Near-infrared (NIR) light, used by IR depth cameras, typically ranges from 850 nm to 940 nm and can penetrate water to a depth of a few millimeters to centimeters. This limited penetration depth is sufficient because typical water blobs on a surface usually have a height of just a few millimeters. We

tested our typing application with our pilot users under various water conditions, ranging from multiple small droplets to a thin puddle on the touch surface. No significant performance differences were observed across these conditions, indicating the potential usage of **HaloTouch** under wet surface scenarios.

5 Evaluation

5.1 Participants and Setup

We recruited 13 participants (6 male and 7 female) from the local university after obtaining approval from our institution's ethics review board (ID: H19-02782-A010). The data from one participant was excluded from the analysis due to non-compliance with the study protocol during the typing task, and we report findings based on the remaining 12 valid participants. The 12 participants have an average age of 25.2 (SD=5.1) with skin types ranging from II to VI on the Fitzpatrick scale [19]. 9 of the 12 people were dominant in their right hand. All reported they use mobile devices to type regularly. 11 out of 12 people reported that they had used AR/VR devices like Vision Pro, HoloLens, or Quest in the past.

As shown in Figure 7, we mounted our Kinect Azure depth camera 0.5 m above the table using extension arms. The projector is mounted right next to the Kinect Azure depth camera to provide projections on the table. An additional side camera is positioned at table height to capture ground truth data for all touch events from a side view. We manually defined a line in the side camera's field of view to align with the surface level, and used Mediapipe's hand tracking to identify fingertip positions. A ground-truth touch event is detected when the fingertip collides with this defined line, enabling robust touch detection and hover distance detection with accuracy of up to 1.2 mm.

The ground truth data accuracy accounts for hand-tracking jitter. The camera's precision is 0.54 mm per pixel, determined by measuring line lengths in the camera view. Jitter analysis, based on the standard deviation of mean positions over 5 seconds, showed an average of 1.26 pixels, corresponding to a 0.68 mm impact on ground truth accuracy.

Throughout the experiment, participants were instructed to extend their index finger while keeping others near the palm, but could adjust clenching, wrist position, and finger posture. Extending multiple fingers, such as the middle or thumb, affected the Halo effect due to interference. While it is possible to model the dynamic interactions between multiple fingers, this study focuses on single-finger interactions for simplicity.

5.2 User Study Protocol

5.2.1 Task One: Target Pointing. Sixteen target points were arranged in a 4×4 grid, evenly spaced within a 20×20 cm square area. One point would appear at a time within this area, and participants were instructed to touch it with a 1 second pause between each point. If a touch was registered within a 2.5 cm radius of the target point, the point would be recorded, disappear, and the next point would appear. If no touch was detected, or if the touch occurred outside the 2.5 cm radius, the current point would remain visible. In such cases, participants were asked to retry touching the point with a different finger angle or force. Each participant completed two sets of regular touches (with touch durations between 0.3s and 0.6s)

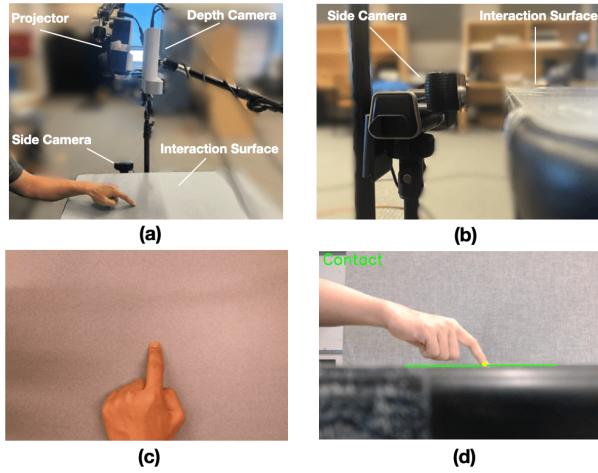


Figure 7: Study Scene Setup: (a) Full setup overview (b) Side-camera aligned with the interaction surface height (c) depth camera RGB view (d) side-camera view

and two sets of swift touches (with touch durations between 0.1s and 0.3s) across five different materials (paper, wood, leather sleeve (suede), plastic, and foam). The points were presented in a random order. Prior to the main test round, participants practiced one set of regular touches and one set of swift touches to get familiar with the protocol, which we do not include in our analysis.

5.2.2 Task Two: Proximity and Pressure Sensing. We 3D printed small cylinders of varying heights: 5 mm, 7 mm, 10 mm, 12 mm, and 15 mm. Participants were provided with one cylinder at a time and instructed to place it under their index finger in a relaxed posture. The hover distance was then recorded from the system. Each participant performed 5 heights \times 2 repetitions: 10 trials in total.

An iPhone 6 was used in the setup to provide ground truth (0 to 385 g) for pressure force measurements using its 3D touch feature. As infrared light is directly reflected off phone screens, preventing any multipath interference, a layer of regular paper was taped over half of the iPhone 6 screen to enable valid readings by our system. The other half of the screen displayed a normalized force reading (0 to 1). Participants were asked to touch the paper-covered half of the screen and adjust their force to five different levels (0.1, 0.25, 0.5, 0.75, 1) according to the display on the iPhone 6. The output value from our system was recorded for each level. Each participant performed 5 pressure levels \times 2 repetitions = 10 trials.

5.2.3 Task Three: Virtual Keyboard Typing. A virtual keyboard is presented on the touch surface for the participants (Figure 13(a)). We used a standard QWERTY keyboard layout, and a standard square key size of 16 mm for our virtual keyboard but with only letters. On top of the keyboard, participants have a reference sentence that they were asked to follow for typing, with their actual input sentence displayed below. The reference sentences were randomly sampled from the widely used Mackenzie phrase set, which contains 500 phrases [42]. Participants can also view their current sentence level

accuracy and words per minute (WPM), so that they can adhere to the different typing goals. We played typing sound feedback at each click for better training and experience, per [41].

We ran a 10-minute warm up session for each participant after a brief introduction of the keyboard elements. In each test round, we asked the participants to type on the virtual keyboard across three tasks, described as follows:

- T1 - Prioritize accuracy: type with more caution to make sure you achieve high accuracy
- T2 - Balance accuracy and speed: type fast like you would on your mobile device, but try to maintain an accuracy of 90%.
- T3 - Prioritize speed: read the reference sentence first, plan out the finger placement trajectory and then type as fast as possible.

For each task we asked the participants to type 8 sentences, and we recorded the average typing accuracy and WPM at the end of each task. We ran N=2 repeats for each task. We note while running the test round, we asked the participants not to use the “delete” key unless it is a mistake that is due to their own behavior. This protocol yields data that avoids user bias towards higher typing accuracy, better reflecting the user true typing input statistics towards the system.

For comparison, we ran one round of the same 8 sentences typed on an Apple iPad Pro (11 inches 2nd Gen) and on a Microsoft HoloLens 2 for each participant. The keyboard (Figure 13(b)) follows exactly the same layout and size implemented as a web application. On HoloLens, the keyboard is implemented using crossing buttons, similar to the default HoloLens 2 keyboard.

5.3 Results

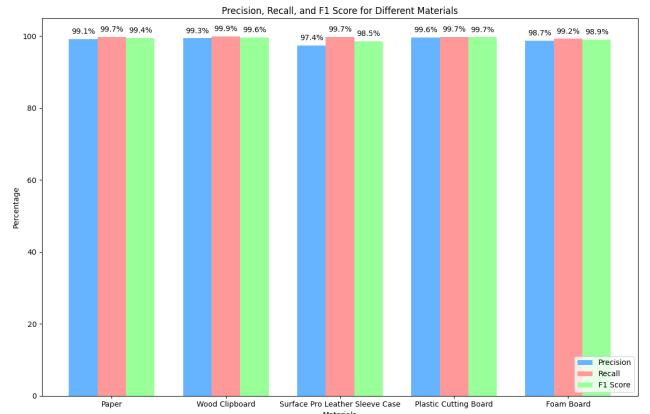


Figure 8: Precision, Recall, and F1 Score for touch detection across five materials: paper, wood, leather sleeve (suede), plastic, and foam in our study.

5.3.1 Touch Down Accuracy. Across 12 participants, we obtained 3840 test touch points. Of these, 13 reported no touch contact (0.34%), 45 reported more than one touch point (1.17%). We report an average precision of 98.8%, recall of 99.7% and F1 score

of 99.2% among all the data points. We show the precision, recall, and F1 score for each material in Figure 8. We see no significant difference ($p>0.01$, ANOVA with Tukey HSD) on the false negative touch counts among material pairs. However, we see significant difference ($p<0.01$, ANOVA with Tukey HSD) on the false positive counts between suede and plastic, and significant difference among other material pairs. We see no significant difference ($p>0.1$, ANOVA) on precision between regular touches and swift touches. Outside of the 3840 points, our system detected 31 points which are outside of the reference point radius.

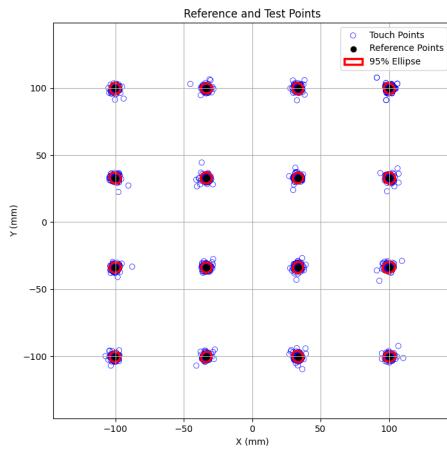


Figure 9: Scatter-plot of all touch points in task one, plotted with 95% confidence ellipses.

5.3.2 Spatial Accuracy. We followed the same analysis procedure as MRTouch [64]. We experienced a similar systematic offset of all received touch points, which is 7 mm to the left. The shift is constant across all materials and users. We subtracted the global average offset from all touch points for the following analysis. We also removed 61 outlier points (1.6%) which lay more than three standard deviations from the reference point. These outliers primarily resulted from user errors and inaccuracies. For instance, some users sped up the point-clicking process after becoming familiar with the task, occasionally failing to lift their fingers fully before clicking the next point. This behavior introduced inconsistencies, which were removed to maintain the integrity and accuracy of the dataset. Across all materials and all participants, we report a global mean Euclidean error of 5.5 mm ($SD = 0.69$).

We see no significant difference between the regular touch and the swift touch conditions ($p>0.01$, ANOVA). Pair-wise Tukey HSD also shows no significant difference ($p>0.01$) among the 5 materials. We plot the 95% confidence ellipses for all the received points in Figure 9. On average, a 16 mm diameter button would capture 95% of touches, which provides additional support for the size selection in our virtual keyboard implementation.

5.3.3 Latency. Our pipeline has an overall mean motion-to-graphics latency of 150 ms, measured using an external high FPS camera.

This matches well with the theoretical latency of 146 ms to 186 ms, composed primarily of camera buffer image loading (33 ms), media pipeline hand tracking (55~85 ms), the rest of the processing pipeline(25~35 ms), and projector delay (33 ms).

5.3.4 Touch Point Threshold. We measure our touch point threshold by averaging all the vertical travel distance of detected fingertips from side camera for each valid key input. We make the assumption that for each key input, the distance of finger traveling down to touch the key and traveling up before aiming for the next key is the same. Our system has a touch point threshold of 4.97 mm, averaging across all participants in the typing application.

Participant	Pressure MAE (%)	Hover Distance MAE (mm)
1	14.36 ± 11.37	1.53 ± 0.61
2	21.56 ± 14.31	1.35 ± 0.28
3	21.16 ± 15.84	2.15 ± 0.27
4	19.00 ± 6.73	4.30 ± 0.38
5	24.84 ± 17.82	1.55 ± 1.35
6	15.60 ± 11.14	2.20 ± 1.27
7	13.08 ± 3.28	4.99 ± 1.75
8	12.32 ± 8.20	3.90 ± 2.41
9	8.40 ± 6.62	1.99 ± 0.69
10	11.28 ± 7.07	1.96 ± 0.74
11	34.80 ± 18.22	2.85 ± 0.64
12	28.84 ± 12.67	4.73 ± 1.46
Average	18.77 ± 11.11	2.81 ± 0.99

Figure 10: Participants pressure MAE and hover distance MAE breakdown

5.3.5 Proximity Accuracy. We show the overall results per participant in Figure 10. We report an average hover distance error of 2.81 ± 0.99 mm. Kruskal-Wallis tests indicated no significant differences between the distance targets ($H = 5.42$, $df = 5$, $p > 0.001$), but indicated significant differences between participants ($H = 34.28$, $df = 12$, $p < 0.001$).

5.3.6 Pressure Accuracy. We show the overall results per participant in Figure 10. We report an average pressure error of $18.77 \pm 11.11\%$. A Kruskal-Wallis test indicated no significant differences between the pressure targets ($H = 8.69$, $df = 5$, $p > 0.001$), and also indicated no significant differences between participants ($H = 28.61$, $df = 12$, $p > 0.001$).

5.3.7 Typing Performance. To evaluate typing performance, we measured both text entry speed and accuracy. Text entry rate (in WPM) was calculated by taking the time difference between the first and last keystrokes for each phrase. For sentence-level accuracy, we split each input sentence into words and compared them to their corresponding reference words at the character level, then reported the average accuracy per participant. We show the task-wise typing accuracy and speed comparison along with the AR and mobile device typing performance in Figure 11.

To present a more intuitive measure combining typing speed and accuracy, we calculated Adjusted Words Per Minute (AWPM), where $AWPM = WPM \times \text{accuracy rate}$. Unlike raw WPM, which only considers the number of words typed in a minute, AWPM considers the error rate as well, providing a more accurate reflection of

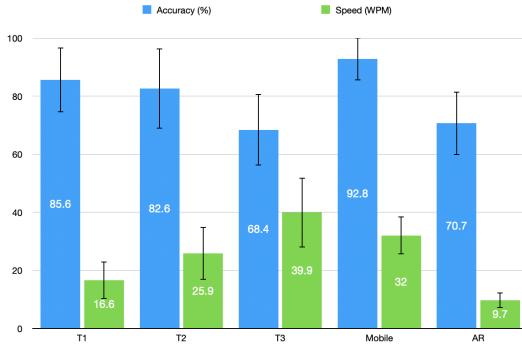


Figure 11: Typing accuracy and speed (WPM) under five conditions: T1, T2, T3, Mobile, and AR



Figure 12: Typing speeds (AWPM) for 12 participants across five conditions. The color gradient visualizes performance differences, with Mobile generally yielding higher speeds and AR having the lowest

typing proficiency. We show the AWPM for each user in Figure 12. Additional metrics for text entry including Uncorrected Error Rate (UER) and Corrected Error Rate (CER) (using definition by [68]) can be found in Appendix C. Using the ground truth data provided from the side camera, we calculated the fingertip-to-surface hover distance when no touch is detected during each typing round for each participant, and produced average participant hover distances for each typing task (T1, T2, T3). We perform Pearson correlation test between AWPM and average hover distance among all participants and found a strong positive correlation (Pearson correlation coefficient = 0.715, p-value < 0.01) for T3, a strong positive correlation for T2 (Pearson correlation coefficient = 0.698, p-value < 0.05), and an insignificantly negative correlation for T1 (Pearson correlation coefficient = -0.197, p-value > 0.05).



Figure 13: Typing Application: (a) Virtual keyboard in HaloTouch projected on wood surface (b) Mobile keyboard with same layout in IPad

6 Example Applications

6.1 Gesture Augmented Virtual Drumming

The HaloTouch system's proximity detection capability allows it to accurately sense the hover distance of the user's fingertip above a surface. This feature is utilized in our virtual drumming app, where users can engage in an expressive drumming experience that goes beyond simple tapping. The app leverages both hands of the user to create a dynamic and rich interactive musical environment.

With one hand, users can play the virtual drum by tapping on any surface, producing realistic drum sounds. Meanwhile, the other hand can be used to hover at various heights above the surface. The distance of this hovering hand from the surface is continuously detected by **HaloTouch**, allowing it to modulate the characteristics of the drum sound in real time. For instance, hovering the hand closer to the surface can enhance the lower frequencies of the drum sound, creating a deeper, bass-like tone, while hovering higher could emphasize higher frequencies, generating sharper, snappier sounds. This modulation effectively alters the harmonic content of the drum sound, creating a wide range of expressive possibilities.

6.2 Pressure Sensitive Painting

The **HaloTouch** system's pressure detection capability enables it to sense the amount of force a user applies on a surface. This feature is utilized in our painting application to provide a more natural and dynamic drawing experience by allowing users to modulate the thickness and style of brush strokes based on varying pressure levels, akin to using traditional art tools like brushes, pens, or calligraphy brushes.

In this application, the amount of pressure detected by **HaloTouch** is seamlessly mapped to stroke attributes such as width, opacity, and texture. Light pressure results in fine, delicate lines suitable for detailing and sketching, while heavier pressure creates broader, more pronounced strokes ideal for bold expressions or filling large areas. This enables users to achieve natural calligraphy effects and provides more freedom and flexibility compared to traditional fixed-stroke painting tools.

The app enhances the painting experience by allowing for smooth transitions between different stroke types within a single brush, enabling artists to create varied textures and effects without having to switch tools frequently. For example, an artist can start with

a light stroke to outline a shape and then gradually increase the pressure to add depth, shading, or emphasis—all within the same continuous motion. This mimics the fluidity of traditional painting techniques, where artists control their brush or pen strokes by applying varying amounts of pressure.

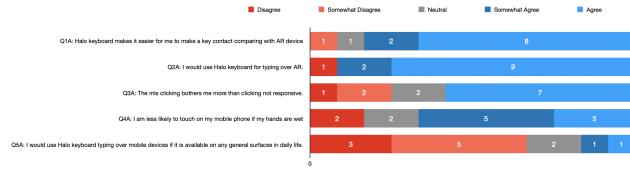


Figure 14: Results from the 5-point Likert scale questions

7 Discussion and Future Work

7.1 Technical Metrics In Comparison with Other Systems

HaloTouch is a comprehensive sensing system that enhances touch interaction on general surfaces by achieving high touch-down accuracy, spatial accuracy, touch point threshold and no instrumentation. In this section, we discuss the key advantages of **HaloTouch** compared to existing systems, focusing on its touch point threshold, versatility, and ability to support diverse input interactions.

Firstly, **HaloTouch** achieves high touch-down accuracy (>99.0%), comparable to systems like Electrick[69], ShadowTouch[39], and Direct[64], and provides high spatial accuracy (<10 mm), similar to Electrick[69], Direct[64], and FarOut[56]. While **HaloTouch** operates with slightly higher latency than some other systems, our typing application demonstrates that a latency of 150 ms does not noticeably impact user experience or performance.

To further enable rapid and accurate touch input, minimizing the touch point threshold is critical. **HaloTouch** achieves a remarkably low touch point threshold of 4.97 mm, outperforming other vision-based approaches such as TapLight[58], Direct[64], FarOut[56], and Tripad[16]. We explain the causes of false positives and false negatives in virtual keyboard typing by discussing the concept of the touch point threshold. Imagine that touching a virtual key is like pressing a physical button with a certain “height” that must be fully pressed down to register a valid touch. The touch point thresholds defines this virtual height. A higher touch point thresholds requires more effort from users to transition from one button to another. In a virtual keyboard, however, without the tactile feedback of physical keys, users often cross into adjacent keys without resetting to the required touch point thresholds. This misalignment is a major cause of false positives or false negatives, depending on the specific threshold settings. Capacitive touch screens, for instance, effectively have a zero touch point thresholds, allowing users to transition smoothly between keys without restrictions. The positive correlation between touch point thresholds and typing performance in T2 and T3 also indicates that users who are used to lift fingers up higher while typing are able to accommodate for systems with a higher touch point threshold. Although systems like DIRECT [64], TapLight [58], and Tripad [16] excel in touch-down

and spatial accuracy, their high touch point thresholds can lead to a high false positive rate during fast interactions. In contrast, while maintaining high touch-down accuracy and spatial accuracy, **HaloTouch**'s low touch point thresholds significantly enhances input speed and accuracy, particularly in rapid typing scenarios.

Another key advantage of **HaloTouch** is its ability to operate on most general materials without requiring surface instrumentation or additional sensors attached to the user. These features are vital for achieving true ad hoc sensing without limiting the user's freedom. In contrast, other systems with high material compatibility and low instrumentation requirements often compromise on spatial accuracy (e.g., Shi et al. [57]) or have a higher touch point threshold (e.g., Tripad [16]), making rapid and accurate touch interactions difficult to support.

Moreover, **HaloTouch**'s capability to detect pressure and proximity enables a richer input interaction space. This includes in-air gestures, pressure-modulated inputs as demonstrated in our example applications, and the combination of hover, touch, and pressure—all using only a commercially available depth camera. With these capabilities, **HaloTouch** extends the interaction modalities on general surfaces, offering a versatile and user-friendly solution for touch-based applications.

7.2 Learning Effect & User Experience

We observed a strong learning effect in the keyboard typing study, which can be explained by two main causes. First, most users are accustomed to typing with their thumbs on mobile devices. While some users reported using their index finger for typing on larger devices like iPads, they do so less frequently. Given that typing performance in T3 for the Halo keyboard was nearly on par with that on mobile devices, it suggests that users can adapt to typing with their index finger after some training and practice. Second, typing on flat surfaces lacks the haptic feedback that users rely on. Although we provided sound feedback when a key was clicked, some participants reported difficulty in knowing whether they had successfully touched a key during the warmup round. For instance, P7 mentioned, “I got really tired in the learning round where I had to constantly make sure if I made contact with the key or not.” Similarly, P9 suggested, “I would enjoy the typing experience even more if the keyboard gives me a vibration.”

Users also faced challenges in estimating how far they needed to reach out in the air to make a key click in HoloLens. P5 described the experience as, “It is hard to click on the key in AR because I cannot feel where the key is relative to my hand.” Conversely, P2 expressed a preference for AR, stating, “I really like how I can see my finger cursor is highlighted in AR, and that helps me know when I am clicking a key.” P12 had mixed feelings: “The AR keyboard requires me to move my whole arm to type, while the Halo keyboard is easier to click on, but the mis-clicks slow down my typing speed.” Notably, 10 out of 12 users (P1, P3-P11) felt that the Halo keyboard made it easier to make key contact compared to the AR device (Q1A).

We noticed that users who were more confident tended to type faster, while those who were more cautious typed slower. Specifically, users who experienced more false positives during the warmup round tended to type more cautiously and slowly in the experiment

round. In contrast, users who typed comfortably in the warmup round often ended up typing even faster in the experiment round. Our implementation of filtering out artifacts based on finger movement speed favors users who type more quickly.

7.3 Privacy & Trust

Almost all users (11 out of 12) preferred typing with the Halo keyboard over the AR keyboard (Q2A). However, 75% of them still favored using their mobile phones, even if the Halo keyboard were available on any general surface in daily life (Q5A). P9 highlighted a critical issue regarding privacy: “Halo keyboard types quite nicely for me, and it would be really cool to have this kind of keyboard everywhere, but I still like my phone better because I don’t want others to see what I am doing.” This sentiment underscores the ongoing trade-off between privacy and convenience. While we believe that **HaloTouch** could offer significant convenience if integrated into everyday life, preserving user privacy remains essential. A potential future work to address this concern could involve deploying **HaloTouch** onto AR headsets. While our system currently works on Kinect Azure, we have confirmed the presence of the Halo effect at varying surfaces and heights (up to 1 m) using HoloLens 2 IR camera feed, validating our working theory. The primary challenge for deployment would involve optimizing for a moving IR light source coupled with a moving plane, ensuring consistent and reliable Halo effect signal strength while maintaining user privacy.

We attribute the lack of correlation between T1 and AWPM to participants’ behavior during experiments. In T1, the emphasis on accuracy may have led participants to prioritize precision over trust-building, suppressing natural typing habits. In T2 and T3, relaxed accuracy requirements likely encouraged more natural typing, improving performance as discussed in Section 7.1.

Beyond technical feasibility, P8 raised another point: “I prefer my phone because I am more comfortable interacting with my own belongings.” This touches on the broader issue of trust between humans and computers, which remains an open research area. Although this paper focuses on enabling new interaction techniques, future research could explore creating an environment where users feel comfortable interacting with any passive surface, much like they do with their personal devices today. This raises new research questions: Can people develop trust in an object they already own if it is augmented with input techniques? Can the **HaloTouch** technique encourage people to trust their environment more over time?

7.4 Sensing Range and Calibration

Our study setup features a sensing distance of 0.5 meters and no yaw or pitch angle on the camera. With our current setup, we support an interaction area of up to 30 cm × 20 cm with index fingertips at a camera distance of 0.5 meters. Our characterization results in Figure 6, 17, and 18 show that the performance of the touch measurement and pressure measurement is not heavily impacted by distance and angle, whereas the resolution of the proximity can be impaired by the distance and the angle of the camera placement. Particularly, the Halo effect strength for proximity falls below 50% at distance of 80 cm, yaw angle of 25 degrees, and pitch angle of 15 degrees. The Halo effect strength for touch input is maintained above 75%

up to distance of 100 cm, yaw angle of 50 degrees, and pitch angle of 45 degrees. The Halo effect strength for pressure is maintained above 75% for up to distance of 70 cm, yaw angle of 35 degrees, and pitch angle of 45 degrees. The Halo effect strength for pressure is amplified between pitch angle -35 degrees to 0 degrees.

The extent to which the Halo effect becomes too weak to support reliable touch detection also depends on factors such as the properties of the surface material. For example, our study found that foam board generates significantly less Halo effect compared to other materials, while the leather sleeve produces a much stronger Halo effect.

Despite the comprehensive sensing capabilities that **HaloTouch** offers, a 20-second calibration is required for each user to accurately capture the finger response across different positions in the interaction area. We additionally characterized the Halo effect given various camera positions. Such characterization can provide extra data to train a more robust algorithm that is free of calibration in the future.

8 Conclusion

We introduced **HaloTouch**, a novel input sensing technique that leverages multipath interference from commercially available depth cameras to enable high-accuracy touch input on a variety of passive surfaces without the need for additional hardware or surface instrumentation. **HaloTouch** excels in its ability to provide comprehensive sensing capabilities, combining both pressure and proximity detection with impressive accuracy across multiple surface types. Our exploration of the Halo Effect has demonstrated the potential of this underutilized phenomenon for surpassing noise thresholds in depth sensors, making it a robust solution for real-world applications. The two-part study validated the technical performance and versatility of **HaloTouch** in supporting rapid, ad hoc interactions in diverse environments, including extreme settings like watery surfaces.

Our work contributes to the growing field of on-world interfaces by providing a flexible, multimodal touch input system that broadens the potential for interaction in uninstrumented environments. Moving forward, **HaloTouch** paves the way for future research in expanding the capabilities of depth cameras and enhancing the seamless integration of digital interactions with the physical world.

Acknowledgments

This work was supported in part by the Natural Science and Engineering Research Council of Canada (NSERC) under Discovery Grant RGPIN-2019-05624 and by Rogers Communications Inc. under the Rogers-UBC Collaborative Research Grant: Augmented and Virtual Reality.

References

- [1] Diar Abdulkarim, Massimiliano Di Luca, Poppy Aves, M. Maaroufi, Sang-Hoon Yeo, R. Miall, Peter Holland, and Joseph Galea. 2023. A methodological framework to assess the accuracy of virtual reality hand-tracking systems: A case study with the Meta Quest 2. *Behavior Research Methods* 56 (02 2023), 1–12. doi:10.3758/s13428-022-02051-8
- [2] Karan Ahuja, Paul Strelci, and Christian Holz. 2021. TouchPose: Hand Pose Prediction, Depth Estimation, and Touch Classification from Capacitive Images. 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, 997–1009. doi:10.1145/3472749.3474801

- [3] Raunaq Bhirangi, Tess Hellebrekers, Carmel Majidi, and Abhinav Gupta. 2021. ReSkin:versatile, replaceable, lasting tactile skins. In *CoRL*.
- [4] Richard A. Bolt. 1980. *"Put-that-there": Voice and gesture at the graphics interface*. Association for Computing Machinery, New York, NY, USA, 262–270. <https://doi.org/10.1145/800250.807503>
- [5] Samarth Brahmbhatt. 2020. *Grasp Contact Between Hand and Object: Capture, Analysis, and Applications*. Ph.D. Dissertation. Georgia Institute of Technology, Atlanta, GA, USA. PhD Thesis.
- [6] Eugenie Brasier, Olivier Chapuis, Nicolas Ferey, Jeanne Vezien, and Caroline Appert. 2020. ARPads: Mid-air Indirect Input for Augmented Reality. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 332–343. doi:10.1109/ISMAR50242.2020.00060
- [7] Leah Buechley, David Mellis, Hannah Perner-Wilson, Emily Lovell, and Bonifaz Kaufmann. 2010. Living wall: programmable wallpaper for interactive spaces. In *Proceedings of the 18th ACM International Conference on Multimedia (Firenze, Italy) (MM '10)*. Association for Computing Machinery, New York, NY, USA, 1401–1402. doi:10.1145/1873951.1874226
- [8] Arturo Cadena, Rubén Carvajal, Bruno Guamán, Roger Granda, Enrique Peláez, and Katherine Chiluiza. 2016. Fingertip detection approach on depth image sequences for interactive projection system. In *2016 IEEE Ecuador Technical Chapters Meeting (ETCM)*, 1–6. doi:10.1109/ETCM.2016.7750827
- [9] Nutan Chen, Göran Westling, Benoni B. Edin, and Patrick van der Smagt. 2019. Estimating Fingertip Forces, Torques, and Local Curvatures from Fingernail Images. *Robotica* 38 (2019), 1242 – 1262. <https://api.semanticscholar.org/CorpusID:202565726>
- [10] Xiang 'Anthony' Chen, Julia Schwarz, Chris Harrison, Jennifer Mankoff, and Scott E. Hudson. 2014. Air+touch: interweaving touch & in-air gestures. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (Honolulu, Hawaii, USA) (UIST '14)*. Association for Computing Machinery, New York, NY, USA, 519–525. doi:10.1145/2642918.2647392
- [11] Yi Fei Cheng, Tiffany Luong, Andreas Rene Fender, Paul Strel, and Christian Holz. 2022. ComforTable User Interfaces: Surfaces Reduce Input Error, Time, and Exertion for Tabletop and Mid-air User Interfaces. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 150–159. doi:10.1109/ISMAR55827.2022.00029
- [12] Frederick Choi, Sven Mayer, and Chris Harrison. 2021. 3D Hand Pose Estimation on Conventional Capacitive Touchscreens. In *Proceedings of the 23rd International Conference on Mobile Human-Computer Interaction (Toulouse & Virtual, France) (MobileHCI '21)*. Association for Computing Machinery, New York, NY, USA, Article 3, 13 pages. doi:10.1145/3447526.3472045
- [13] R. L. Cook and K. E. Torrance. 1982. A Reflectance Model for Computer Graphics. *ACM Trans. Graph.* 1, 1 (jan 1982), 7–24. doi:10.1145/357290.357293
- [14] Gergo Csonka, Muhammad Khalid, Husnain Rafiq, and Yasir Ali. 2023. AI-Based Hand Gesture Recognition Through Camera on Robot. In *2023 International Conference on Frontiers of Information Technology (FIT)*. 256–261. doi:10.1109/FIT60620.2023.00054
- [15] Rhett Dobinson, Marc Teyssier, Jürgen Steimle, and Bruno Fruchard. 2022. Micro-Press: Detecting Pressure and Hover Distance in Thumb-to-Finger Interactions. In *Proceedings of the 2022 ACM Symposium on Spatial User Interaction (Online, CA, USA) (SUI '22)*. Association for Computing Machinery, New York, NY, USA, Article 4, 10 pages. doi:10.1145/3565970.3567698
- [16] Camille Dupré, Caroline Appert, Stéphanie Rey, Houssem Saidi, and Emmanuel Pietriga. 2024. TriPad: Touch Input in AR on Ordinary Surfaces with Hand Tracking Only. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (Honolulu, HI, USA) (CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 754, 18 pages. doi:10.1145/3613904.3642323
- [17] Neil Xu Fan and Robert Xiao. 2022. Reducing the Latency of Touch Tracking on Ad-hoc Surfaces. *Proc. ACM Hum.-Comput. Interact.* 6, ISS, Article 577 (nov 2022), 11 pages. doi:10.1145/3567730
- [18] Michal Feigin, Ayush Bhandari, Shahram Izadi, Christoph Rhemann, Mirko Schmidt, and Ramesh Raskar. 2016. Resolving Multipath Interference in Kinect: An Inverse Problem Approach. *IEEE Sensors Journal* 16, 10 (2016), 3419–3427. doi:10.1109/JSEN.2015.2421360
- [19] T. B. Fitzpatrick. 1975. Soleil et peau. *J Med Esthet* 2, 7 (1975), 33–34.
- [20] Google Research. 2023. MediaPipe Hand Landmarker: Real-time Hand Landmark Detection. https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker. Accessed: 2024-09-01.
- [21] P. Grady, J. A. Collins, C. Tang, C. D. Twigg, K. Aneja, J. Hays, and C. C. Kemp. 2024. PressureVision++: Estimating Fingertip Pressure from Diverse RGB Images. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE Computer Society, Los Alamitos, CA, USA, 8683–8693. doi:10.1109/WACV57701.2024.00850
- [22] 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: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VI* (Tel Aviv, Israel). Springer-Verlag, Berlin, Heidelberg, 328–345. doi:10.1007/978-3-031-20068-7_19
- [23] Yizheng Gu, Chun Yu, Zhipeng Li, Weiqi Li, Shuchang Xu, Xiaoying Wei, and Yuanchun Shi. 2019. Accurate and Low-Latency Sensing of Touch Contact on Any Surface with Finger-Worn IMU Sensor. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (New Orleans, LA, USA) (UIST '19)*. Association for Computing Machinery, New York, NY, USA, 1059–1070. doi:10.1145/3332165.3347947
- [24] Anhong Guo, Robert Xiao, and Chris Harrison. 2015. CapAuth: Identifying and Differentiating User Handprints on Commodity Capacitive Touchscreens. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces (Madeira, Portugal) (ITS '15)*. Association for Computing Machinery, New York, NY, USA, 59–62. doi:10.1145/2817721.2817722
- [25] 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. doi:10.1145/2047196.2047255
- [26] 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. doi:10.1145/1449715.1449747
- [27] Nadia Haubner, Ulrich Schwancke, Ralf Dörner, Simon Lehmann, and Johannes Luderschmidt. 2013. Detecting interaction above digital tabletops using a single depth camera. *Mach. Vision Appl.* 24, 8 (nov 2013), 1575–1587. doi:10.1007/s00138-013-0538-5
- [28] Juan David Hincapíe-Ramos, Xiang Guo, Paymahn Moghadasi, and Pourang Irani. 2014. Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Toronto, Ontario, Canada) (CHI '14)*. Association for Computing Machinery, New York, NY, USA, 1063–1072. doi:10.1145/2556288.2557130
- [29] Ken Hinckley, Seongkook Heo, Michel Pahud, Christian Holz, Hrvoje Benko, Abigail Sellen, Richard Banks, Kenton P. O'Hara, Gavin Smyth, and William A.S. Buxton. 2016. Pre-Touch Sensing for Mobile Interaction. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (2016). <https://api.semanticscholar.org/CorpusID:15482351>
- [30] Xincheng Huang, James Riddell, and Robert Xiao. 2023. Virtual Reality Telepresence: 360-Degree Video Streaming with Edge-Compute Assisted Static Foveated Compression. *IEEE Transactions on Visualization and Computer Graphics* 29, 11 (2023), 4525–4534. doi:10.1109/TVCG.2023.3320255
- [31] Xincheng Huang and Robert Xiao. 2024. SurfShare: Lightweight Spatially Consistent Physical Surface and Virtual Replica Sharing with Head-mounted Mixed-Reality. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 7, 4, Article 162 (Jan. 2024), 24 pages. doi:10.1145/3631418
- [32] Xincheng Huang, Michael Yin, Ziyi Xia, and Robert Xiao. 2024. VirtualNexus: Enhancing 360-Degree Video AR/VR Collaboration with Environment Cutouts and Virtual Replicas. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology (Pittsburgh, PA, USA) (UIST '24)*. Association for Computing Machinery, New York, NY, USA, Article 55, 12 pages. doi:10.1145/3654777.3676377
- [33] Wonjun Hwang and Soo-Chul Lim. 2017. Inferring Interaction Force from Visual Information without Using Physical Force Sensors. *Sensors (Basel, Switzerland)* 17 (2017). <https://api.semanticscholar.org/CorpusID:12487487>
- [34] Hiroshi Ishii, Craig Wisneski, Julian Orbanes, Ben Chun, and Joe Paradiso. 1999. PingPongPlus: design of an athletic-tangible interface for computer-supported cooperative play. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Pittsburgh, Pennsylvania, USA) (CHI '99)*. Association for Computing Machinery, New York, NY, USA, 394–401. doi:10.1145/302979.303115
- [35] Ragui Kamel. 1990. Guest Editor's Introduction: Voice in Computing. *Computer* 23, 8 (aug 1990), 8–9.
- [36] Hongki Kim, Seung-Gun Lee, and Kwang-Seok Yun. 2011. Capacitive tactile sensor array for touch screen application. *Sensors and Actuators A-physical* 165 (2011), 2–7. <https://api.semanticscholar.org/CorpusID:109271148>
- [37] Daniel Kurz. 2014. Thermal touch: Thermography-enabled everywhere touch interfaces for mobile augmented reality applications. In *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 9–16. doi:10.1109/ISMAR.2014.6948403
- [38] R. Lange and P. Seitz. 2001. Solid-state time-of-flight range camera. *IEEE Journal of Quantum Electronics* 37, 3 (2001), 390–397. doi:10.1109/3.910448
- [39] Chen Liang, Xutong Wang, Zisu Li, Chi Hsia, Mingming Fan, Chun Yu, and Yuanchun Shi. 2023. ShadowTouch: Enabling Free-Form Touch-Based Hand-to-Surface Interaction with Wrist-Mounted Illuminant by Shadow Projection. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (San Francisco, CA, USA) (UIST '23)*. Association for Computing Machinery, New York, NY, USA, Article 27, 14 pages. doi:10.1145/3586183.3606785
- [40] Robert W. Lindeman, John L. Sibert, and James K. Hahn. 1999. Towards usable VR: an empirical study of user interfaces for immersive virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (Pittsburgh, Pennsylvania, USA) (CHI '99)*. Association for Computing Machinery,

- New York, NY, USA, 64–71. doi:10.1145/302979.302995
- [41] Zhaoxuan Ma, Darren Edge, Leah Findlater, and Hong Z. Tan. 2015. Haptic keyclick feedback improves typing speed and reduces typing errors on a flat keyboard. In *2015 IEEE World Haptics Conference (WHC)* (2015), 220–227. <https://api.semanticscholar.org/CorpusID:15523816>
- [42] I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase sets for evaluating text entry techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA) (*CHI EA '03*). Association for Computing Machinery, New York, NY, USA, 754–755. doi:10.1145/765891.765971
- [43] Stephen A. Mascaro and Haruhiko Asada. 2004. Measurement of finger posture and three-axis fingertip touch force using fingernail sensors. *IEEE Transactions on Robotics and Automation* 20 (2004), 26–35. <https://api.semanticscholar.org/CorpusID:7842621>
- [44] Stephen A. Mascaro and H. Harry Asada. 2001. Photoplethysmograph fingernail sensors for measuring finger forces without haptic obstruction. *IEEE Trans. Robotics Autom.* 17 (2001), 698–708. <https://api.semanticscholar.org/CorpusID:17720714>
- [45] Fabrice Matulic, Taiga Kashima, Deniz Beker, Daichi Suzuo, Hiroshi Fujiwara, and Daniel Vogel. 2023. Above-Screen Fingertip Tracking with a Phone in Virtual Reality. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) (*CHI EA '23*). Association for Computing Machinery, New York, NY, USA, Article 18, 7 pages. doi:10.1145/3544549.3585728
- [46] Daniel Medeiros, Graham Wilson, Mark McGill, and Stephen Anthony Brewster. 2023. The Benefits of Passive Haptics and Perceptual Manipulation for Extended Reality Interactions in Constrained Passenger Spaces. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) (*CHI '23*). Association for Computing Machinery, New York, NY, USA, Article 232, 19 pages. doi:10.1145/3544548.3581079
- [47] M. Meier, P. Streli, A. Fender, and C. Holz. 2021. TapID: Rapid Touch Interaction in Virtual Reality using Wearable Sensing. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE Computer Society, Los Alamitos, CA, USA, 519–528. doi:10.1109/VR50410.2021.00076
- [48] Microsoft Corporation. 2019. Azure Kinect DK. <https://learn.microsoft.com/en-us/previous-versions/azure/kinect-dk/>. Accessed: 2024-09-01.
- [49] Vimal Mollyn and 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* (Pittsburgh, PA, USA) (*UIST '24*). Association for Computing Machinery, New York, NY, USA, Article 69, 11 pages. doi:10.1145/3654777.3676455
- [50] Ju Young Oh, Ji-Hyung Park, and Jung-Min Park. 2020. FingerTouch: Touch Interaction Using a Fingernail-Mounted Sensor on a Head-Mounted Display for Augmented Reality. *IEEE Access* 8 (2020), 101192–101208. doi:10.1109/ACCESS.2020.2997972
- [51] J.A. Paradiso, Che King Leo, N. Checka, and Kaijen Hsiao. 2002. Passive acoustic sensing for tracking knocks atop large interactive displays. In *SENSORS, 2002 IEEE*, Vol. 1, 521–527 vol.1. doi:10.1109/ICSENS.2002.1037150
- [52] Siyou 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. doi:10.1145/3526113.3545622
- [53] Tu-Hoa Pham, Nikolaos Kyriazis, Antonis A. Argyros, and Abderrahmane Kheddar. 2018. Hand-Object Contact Force Estimation from Markerless Visual Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40 (2018), 2883–2896. <https://api.semanticscholar.org/CorpusID:51612933>
- [54] Christian Rendl, Patrick Greindl, Kathrin Probst, Martin Behrens, and Michael Haller. 2014. Pressstures: exploring pressure-sensitive multi-touch gestures on trackpads. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) (*CHI '14*). Association for Computing Machinery, New York, NY, USA, 431–434. doi:10.1145/2556288.2557146
- [55] Christopher Schmandt and Eric A. Hulteen. 1982. The intelligent voice-interactive interface. In *Proceedings of the 1982 Conference on Human Factors in Computing Systems* (Gaithersburg, Maryland, USA) (*CHI '82*). Association for Computing Machinery, New York, NY, USA, 363–366. doi:10.1145/800049.801812
- [56] Vivian Shen, James Spann, and Chris Harrison. 2021. FarOut Touch: Extending the Range of ad hoc Touch Sensing with Depth Cameras. In *Proceedings of the 2021 ACM Symposium on Spatial User Interaction (Virtual Event, USA) (SUI '21)*. Association for Computing Machinery, New York, NY, USA, Article 5, 12 pages. doi:10.1145/3485279.3485281
- [57] 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 (jun 2020), 25 pages. doi:10.1145/3397309
- [58] Paul Streli, 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* (San Francisco, CA, USA) (*UIST '23*). Association for Computing Machinery, New York, NY, USA, Article 26, 12 pages. doi:10.1145/3586183.3606749
- [59] Paul Streli, Mark Richardson, Fadi Botros, Shugao Ma, Robert Wang, and Christian Holz. 2024. TouchInsight: Uncertainty-aware Rapid Touch and Text Input for Mixed Reality from Egocentric Vision. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) (*UIST '24*). Association for Computing Machinery, New York, NY, USA, Article 7, 16 pages. doi:10.1145/3654777.3676330
- [60] Radu-Daniel Vatavu. 2023. iFAD Gestures: Understanding Users' Gesture Input Performance with Index-Finger Augmentation Devices. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) (*CHI '23*). Association for Computing Machinery, New York, NY, USA, Article 576, 17 pages. doi:10.1145/3544548.3580928
- [61] Andrew D. Wilson. 2010. Using a depth camera as a touch sensor. In *International Conference on Intelligent Tutoring Systems*. <https://api.semanticscholar.org/CorpusID:1333692>
- [62] Bo Xiao. 2018. On-World Computing: Enabling Interaction on Everyday Surfaces. (10 2018). doi:10.1145/R1/7195202.v1
- [63] Robert Xiao, Chris Harrison, and Scott E. Hudson. 2013. WorldKit: rapid and easy creation of ad-hoc interactive applications on everyday surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) (*CHI '13*). Association for Computing Machinery, New York, NY, USA, 879–888. doi:10.1145/2470654.2466113
- [64] Robert Xiao, Scott Hudson, and Chris Harrison. 2016. DIRECT: Making Touch Tracking on Ordinary Surfaces Practical with Hybrid Depth-Infrared Sensing. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces* (Niagara Falls, Ontario, Canada) (*ISS '16*). Association for Computing Machinery, New York, NY, USA, 85–94. doi:10.1145/2992154.2992173
- [65] Robert Xiao, Greg Lew, James Marsanico, Divya Hariharan, Scott Hudson, and Chris Harrison. 2014. Toffee: enabling ad hoc, around-device interaction with acoustic time-of-arrival correlation. In *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices & Services* (Toronto, ON, Canada) (*MobileHCI '14*). Association for Computing Machinery, New York, NY, USA, 67–76. doi:10.1145/2628363.2628383
- [66] Robert Xiao, Julia Schwarz, Nick Throm, Andrew D. Wilson, and Hrvoje Benko. 2018. MRTouch: Adding Touch Input to Head-Mounted Mixed Reality. *IEEE Transactions on Visualization and Computer Graphics* 24, 4 (2018), 1653–1660. doi:10.1109/TVCG.2018.2794222
- [67] Hui-Shyong Yeo, Erwin Wu, Daehwa Kim, Juyoung Lee, Hyung-il Kim, Seo Young Oh, Luna Takagi, Woontack Woo, Hideki Koike, and Aaron John Quigley. 2023. OmniSense: Exploring Novel Input Sensing and Interaction Techniques on Mobile Device with an Omni-Directional Camera. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) (*CHI '23*). Association for Computing Machinery, New York, NY, USA, Article 530, 18 pages. doi:10.1145/3544548.3580747
- [68] Mingrui Ray Zhang and Jacob O. Wobbrock. 2019. Beyond the Input Stream: Making Text Entry Evaluations More Flexible with Transcription Sequences. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) (*UIST '19*). Association for Computing Machinery, New York, NY, USA, 831–842. doi:10.1145/3332165.3347922
- [69] Yang Zhang, Gierad Laput, and Chris Harrison. 2017. ElectricR: Low-Cost Touch Sensing Using Electric Field Tomography. 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, 1–14. doi:10.1145/3025453.3025842
- [70] Yang Zhang, Chouchang (Jack) Yang, Scott E. Hudson, Chris Harrison, and Alanson Sample. 2018. Wall++: Room-Scale Interactive and Context-Aware Sensing. 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. doi:10.1145/3173574.3173847
- [71] Yang Zhang, Junhan Zhou, Gierad 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* (San Jose, California, USA) (*CHI '16*). Association for Computing Machinery, New York, NY, USA, 1491–1503. doi:10.1145/2858036.2858082

A Appendix A: Design Space Scoring Table

	Instrumentation	Touch Down Accuracy(%)	Spatial Accuracy (mm)	Latency (ms)	Touch Point Threshold (mm)	Material Compatibility	Pressure	Proximity
Electric	Electrodes around surface	99.5	9.1	3	50	Low (conductive or need coating)	N/A	N/A
ShadowTouch	Wearable (light emitter on wrist) + Headset	99.1	11	20	2	Low (color other than white affects lighting)	N/A	N/A
Gu	Wearable (IMU on finger)	98.61	11	10	5	Medium (unclear if it works on damping/softer material)	N/A	N/A
Shi	Wearable (IMU on finger)	96	11	109	2	High (Specifically evaluated on different materials)	N/A	N/A
Taplight	Headset	95.3	11	50.4	45	Medium (Works on regular desks, unclear if it works on damping/softer materials that affect speckle)	N/A	N/A
Direct	Kinect	99	4.8	45	30	Medium (works on regular desks)	N/A	N/A
Farout	Kinect	96.1	9.8	233	10	Medium (works on regular desks)	N/A	N/A
Micropress	Wearable (IMU)	94.11	N/A	20	0.47	Low (Works only on fingers)	6.71	0.57
HaloTouch	Kinect	99.2	5.5	150	4.97	Medium to High (evaluated on surfaces with various materials)	18.77	2.81
Normalized								
Electric	2	9.95	7.17	10.00	0.00	2	0	0
ShadowTouch	8	9.91	5.92	9.26	9.69	1	0	0
Gu	4	9.86	5.92	9.70	9.09	5	0	0
Shi	4	9.60	5.92	5.39	9.69	10	0	0
Taplight	10	9.53	5.92	7.94	1.01	5	0	0
Direct	10	9.90	10.00	8.17	4.04	5	0	0
Farout	10	9.61	6.71	0.00	8.08	5	0	0
Micropress	4	9.41	0.00	9.26	10.00	1	10	0
HaloTouch	10	9.92	9.54	3.61	9.09	8	6	6

Figure 15: Comparison of Touch Input Systems Across Key Metrics. This table compares various touch input systems based on key performance metrics, including touch-down accuracy, spatial accuracy, latency, touch point threshold, material compatibility, pressure sensing, and proximity sensing.

B Appendix B: Halo Effect Signal Strength

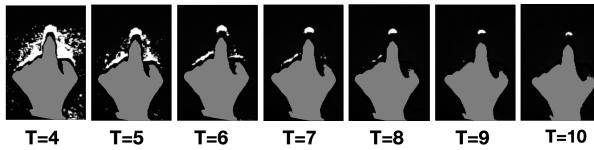


Figure 16: Effect of Threshold Variation on Halo Signal Strength. This figure illustrates the impact of different threshold values (T=4 to T=10) on the detection of the halo effect around a fingertip in depth camera imagery.

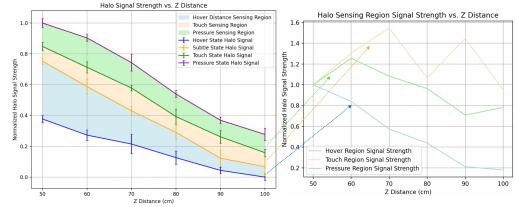


Figure 17: Halo Signal Strength Variation Across Z-Distance. These plots illustrate the relationship between Z-distance (distance from the sensor) and normalized halo signal strength for different interaction states (hover, touch, and pressure). The left plot shows signal strength degradation with increasing Z-distance, with shaded regions indicating different sensing zones. The right plot highlights how signal strength transitions across the halo sensing regions.

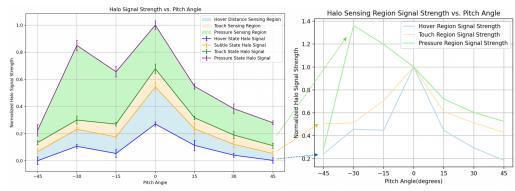


Figure 18: Halo Signal Strength Variation Across Pitch Angles. These plots illustrate the relationship between pitch angle and normalized halo signal strength for different interaction states (hover, touch, and pressure). The left plot shows how signal strength changes with pitch angle, where the shaded regions indicate hover, touch, and pressure sensing zones. The right plot focuses on the transition of signal strength across these sensing regions.

C Appendix C: Typing UER and CER

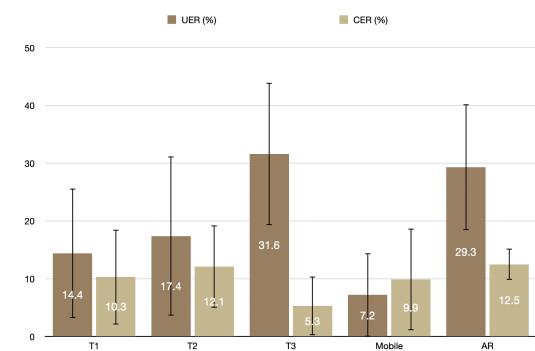


Figure 19: User Error Rate (UER) and Character Error Rate (CER) across different input methods. This bar chart compares UER (%) and CER (%) across five input conditions: T1, T2, T3, Mobile, and AR. Darker bars represent UER, while lighter bars represent CER, with values displayed inside each bar.