Palmpad: Enabling Real-Time Index-to-Palm Touch Interaction with a Single RGB Camera

Zhe He*
Department of Computer Science and Technology
Tsinghua University
Beijing, Beijing, China
hz23@mails.tsinghua.edu.cn

Xiangyang Wang*
Department of Computer Science and Technology
Tsinghua University
Beijing, China
xiangyan20@mails.tsinghua.edu.cn

Yuanchun Shi†
Department of Computer Science and Technology
Tsinghua University
Beijing, China
Qinghai University
Xining, China
shiyc@tsinghua.edu.cn

Chi Hsia
Department of Computer Science and Technology
Tsinghua University
Beijing, China
xq22@mails.tsinghua.edu.cn

Chen Liang
Computational Media and Arts Thrust
The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, Guangdong, China
lliangchenc@163.com

Chun Yu
Department of Computer Science and Technology
Tsinghua University
Beijing, China
chunyu@tsinghua.edu.cn

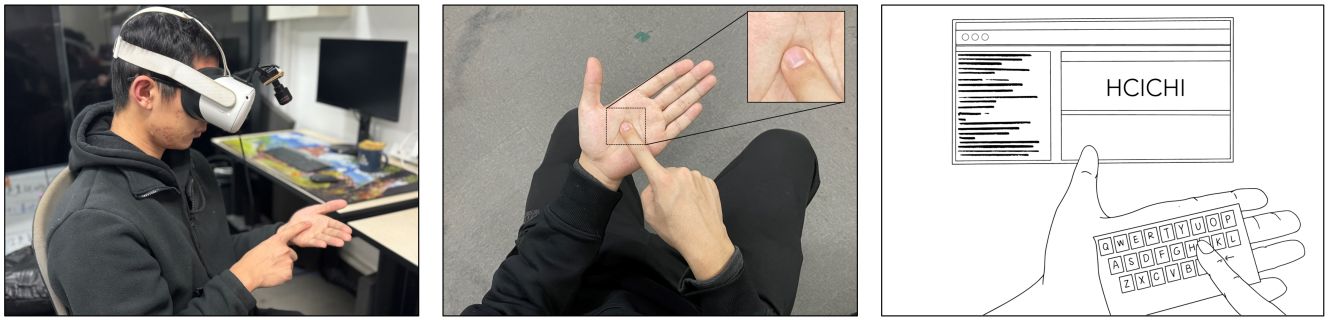


Figure 1: Demonstration of using the PalmPad system to input text directly in a VR scenario on the palm. (Left) A user interacts with his palm with finger taps, captured by a monocular RGB camera integrated into a head-mounted display. (Middle) The picture captured by the head-mount camera and the main feature of the finger used is amplified. (Right) An application scenario, where a virtual keyboard is displayed, allowing for text input via palm-based finger tapping.

Abstract

Index-to-palm interaction plays a crucial role in Mixed Reality(MR) interactions. However, achieving a satisfactory inter-hand interaction experience is challenging with existing vision-based hand tracking technologies, especially in scenarios where only a single camera is available. Therefore, we introduce Palmpad, a novel sensing method utilizing a single RGB camera to detect the touch of an index finger on the opposite palm. Our exploration reveals that the incorporation of optical flow techniques to extract motion information between consecutive frames for the index finger and palm leads to a significant improvement in touch status determination. By doing so, our CNN model achieves 97.0% recognition accuracy and a 96.1% F1 score. In usability evaluation, we compare Palmpad with Quest's inherent hand gesture algorithms. Palmpad not only delivers superior accuracy 95.3% but also reduces operational demands and significantly improves users' willingness and confidence. Palmpad aims to enhance accurate touch detection for lightweight MR devices.

CCS Concepts

• **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**.

Keywords

finger touch, index-to-palm interaction, computer vision

*Both authors contributed equally to this research.
†Corresponding author.

CHI '25, Yokohama, Japan

ACM ISBN 979-8-4007-1394-1/25/04
https://doi.org/10.1145/3706598.3714130

ACM Reference Format:
Zhe He, Xiangyang Wang, Yuanchun Shi, Chi Hsia, Chen Liang, and Chun Yu. 2025. Palmpad: Enabling Real-Time Index-to-Palm Touch Interaction with a Single RGB Camera. In *CHI Conference on Human Factors in Computing Systems (CHI '25), April 26–May 01, 2025, Yokohama, Japan.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3706598.3714130

1 Introduction

In the realm of Mixed Reality (MR) environments, index-to-palm interaction emerges as a convenient mode of interaction [45]. This interaction method treats one palm as a touch surface, with the index finger of the other hand serving as the input tool for clicking or swiping, providing tactile feedback and reducing user fatigue while enhancing input efficiency.

Existing MR devices can be categorized into single-camera and multi-camera systems. Some lightweight MR devices [21, 59] (as well as some wearable camera systems like Ai pin [30]) tend to utilize a single RGB camera, which limits their hand-tracking capabilities, particularly in-depth information. Even with multi-camera MR devices, such as Microsoft Hololens 2 [16] and Meta Quest 2 [19], the depth estimation for index-to-palm interactions is still insufficient to support hand interaction with higher spatial resolution. For example, it is difficult for them to differentiate between an index finger on the palm and an index finger slightly above the palm, which makes them inefficient in detecting touch states in index-to-palm interactions [41]. To solve this problem, some research has opted for the use of additional wearable devices and sensors for detection [15, 45, 79], which may be inconvenient for users. Additionally, it is necessary to detect the timing of clicks to support light/firm/fast/slow click/swipe, which is common in our pilot study and challenging. To solve this problem, previous works [62] suggest (but haven't implemented) incorporating temporal information by concatenating consecutive frames or adding temporal neural networks like LSTM, which is proven to be limited in our study. Other works like [13] and [14] try to predict the pressure between fingers and the surface, which is insufficient to detect the click on the soft and deformable palm.

In this paper, we aim to introduce a highly usable and easily deployable touch state detection solution for index-to-palm interactions using a single RGB camera, which we refer to as "Palmpad". Palmpad can accurately determine the current touch state, supporting various touch gestures such as clicking and swiping, thus providing an always-available virtual keyboard and touchpad with tactile feedback that can be readily deployed on existing MR devices.

We first conducted a data collection experiment involving 16 participants, gathering an index-to-palm dataset with a duration of 211 minutes, which includes two types of actions: index-click-palm and index-swipe-palm. Ground truth calibration was achieved using an AC circuit.

Subsequently, we conducted a detailed analysis of the video data recorded to identify the features that determine the touch state. We observed that the key image features influencing the touch state were primarily located around the right index fingertip, and temporal information was particularly important for click detection. Therefore, we utilized Mediapipe [85] for hand tracking to segment images of the left-hand palm and the right-hand index fingertip. We computed global dense optical flow for consecutive frames to extract temporal information, which significantly improved performance while reducing the parameter scale and the computational load, demonstrating high applicability. Using this data as input, we designed a CNN model with ResNet [26] pre-trained on ImageNet [9] as the backbone and employed fully connected layers for classifying touch and non-touch states. Evaluation experiments demonstrated that this model effectively extracted feature information relevant to index-to-palm interaction, achieving an accuracy of 97.0% and an F1 score of 96.1% in the leave-one-out touch state detection. This level of accuracy surpasses similar work (Acc of 89.1% and F1 score of 85.2% in [62]) in the field. Furthermore, when compared with the approach of replacing optical flow with LSTM, deleting optical flow, or Mediapipe cropping as an ablation study, our model displayed a significant advantage, highlighting the strength of our model in extracting temporal information.

Finally, we conducted a usability experiment, comparing Palmpad's performance to state-of-the-art(SOTA) commercial techniques via two tasks: index-click-palm and index-swipe-palm. Results revealed that Palmpad reduced task completion time by 3.1% while improving accuracy by 2.8% compared with the SOTA commercial techniques. User subjective evaluations indicated that Palmpad garnered significantly higher acceptance and lower perceived burden.

In summary, Palmpad makes three main contributions:

(1) We proposed a method to enable index-to-palm interactions on MR devices using only a single RGB camera. It may be possible to seamlessly integrate this method into existing MR devices by reusing the headset's front-facing camera.

(2) We presented an effective CNN-based index-to-palm touch recognition model that took different levels of image features as input, along with a prototypical implementation of Palmpad. Our model achieved an optimal touch recognition accuracy of 97.0%. User study showed Palmpad outperformed state-of-the-art commercial solutions with a 2.8% improvement in accuracy and a 50% enhancement in spatial resolution.

(3) We explored the extensive application space of Palmpad and implemented a subset of it. Our usability evaluation study validated the high usability of Palmpad compared with the threshold-based approach.

2 RELATED WORKS

2.1 On-body Interface

Recently, there has been increasing attention on utilizing the body as an interface. This approach involves using the human body as a consistently available interface for both input and output, which is considered to have tremendous potential. When users interact with their own bodies by touch, it can unify their cognitive and physical actions [23, 74]. The human body has a large external surface area, most of which is accessible by the hands, implying a vast interactive interface. Furthermore, the unique proprioception of the human body, which involves perceiving the positions of its parts in space, allows for interaction on the body surface without relying on visual attention [25].

In practical interactive scenarios, such input methods have been proven to be effective, including sports applications [70], visually impaired individual applications [54], medical applications [33], as well as XR applications [82].

So far, numerous works have investigated different parts of the human body, including palms [10, 17, 18, 71, 72], fingers [6, 17, 29, 81], nails [34], forearms [47], back of the hand [43], and skin [73]. For example, Omnitouch [22] takes the palm as a clicking interface to expand the user's interactive area, allowing for a broader range of interactions. AI-ON-SKIN [4] utilizes the surface of artificial skin for interaction and computation, which enables tasks such as handwriting and gesture recognition. WatchSense [66] uses a depth sensor embedded in a wearable device to enhance input on the skin. Egotouch [51] achieves hand skin input using just an RGB camera in XR headsets. ClothFace [47] achieves the estimation of human body posture via the mutual perception of the forearm and passive radio frequency signals. KnitDermis [38] takes soft knitting to deliver tactile sensations on the skin, which can conform to underexplored body locations, such as protruded joints and convex body locations. Beyond exploring the possibilities of interaction in different parts of the human body, Ashbrook et al. [2] find that a wrist-mounted system is significantly faster to access than a device stored in a pocket or mounted on the hip.

Our work shares a similar goal of reducing users' burden through proprioception. The index-to-palm interactions, which are designed in Palmpad, enable users to perform touch operations without relying on visual attention. In the meantime, the palm, serving as a flat surface for interaction, can support various types of interactions such as text input, touchpad, and many others, providing a vast interaction space.

2.2 Enhancing Touch Recognition on Unmodified Surface

Researchers in Human-Computer Interaction (HCI) have consistently been exploring methods for touch interaction on unmodified surfaces, which aligns with this study, as we focus on using the palm as an unmodified surface for touch interaction. Therefore, it is necessary to review the works in this field.

The touch on a surface can be divided into several stages. The finger touches down and then maintains contact with the surface. After that, the finger touches up and then remains in a non-contact state [41], which means that touch includes motion information and mechanical information related to the surface. Meanwhile, touch can also generate various accompanying information, such as sound [24], shadows [62], electrical signals [28], and so on. The diverse information generated by the interaction could support distinct interaction space.

Researchers employ various methods to capture diverse information, among which, motion information is frequently detected using IMUs because they are suitable for capturing subtle movements and vibrations, thereby achieving higher sensing accuracy [15]. ActualTouch [63] uses a single nailed-mounted IMU to detect touch. QwertyRing [80] uses an IMU ring worn on the middle phalanx of the index finger, enabling the text entry technique. DualRing [42], which is composed of two IMU rings and a high-frequency AC circuit, is designed to sense rich hand information. Mechanical information is also utilized for touch detection. Paradiso et al. [56] recognize contact interactivity by measuring the position of a knock. WhichFinger [46] relies on vibration sensors attached to each finger to identify touch with low latency. On the aspect of sound, Skinput [25] detects the location of finger taps on the arm and hand by acoustic sensors worn as an armband. AudioTouch [40] attaches two piezo-electric elements as a speaker and microphone on the back of the hand to sense hand gestures. In terms of optical signals, ShadowTouch [41] enhances the shadow during contact and utilizes a camera to detect touch. Agarwal et al. [1] implement Multi-touch Interaction using cameras. StegoType [60] adapts ideas from end-to-end ASR and domain-specific qualities of two-handed typing hand motions to achieve typing on uninstrumented flat surfaces. TouchInsight [68] uses a bivariate Gaussian distribution to represent the location and achieve touch detection from all ten fingers on any surface. Structured Light Speckle [67] leverages structured laser light and egocentric optical sensing to detect touch on discovered physical surfaces. HumTouch [28] takes electrodes attached to the surface of an object to achieve touch position estimation. ElectroRing [37] uses an electrical technique to measure the precise moment of contact and release between the fingertip and the skin. ActiTouch [86] uses the human body as an RF waveguide to achieve touch segmentation.

While the aforementioned methods can be used for touch recognition, each has its own limitations. For instance, detection based on IMU, vibration, or sound focuses on identifying touch events [41], making them ineffective for continuous sliding contact on a surface [15]. Additionally, these methods often introduce additional wearable devices. As for the use of electrical signals, although they can accomplish touch state detection and even position estimation [28], the approaches of introducing electrodes are challenging to apply to different surfaces and human body. Moreover, the long-term effects of passing electric current through the human body have not been fully evaluated. Existing optical methods also have limitations. They either require depth cameras [22] or depend on additional light sources [41] to enhance shadows. Our work adopts only optical sensing without the need for any other additional wearable devices. Given the camera setup being used, it is suitable to reuse the front-facing cameras on existing VR headsets as sensors and specialized features of the palm, which allows us to enhance touch detection, achieving effective results without the need for extra devices.

2.3 Mining Implicit Features from Vision-based Tracking and Recognition

Hand detection and tracking based on vision have been drawing researchers' wide attention. Researchers have employed visual methods to achieve numerous downstream tasks related to hand such as palm print recognition [31, 36], palm vein recognition [57, 64, 78], hand posture reconstruction [35], gesture recognition [52, 55, 58], and hand tracking [7].

In general, two main approaches can be implemented for hand detection and tracking based on vision [49]. One of them is taking markers as an aid. Han et al. [20] designed a glove equipped with multiple markers to assist in hand tracking and identification. Zaman et al. [83] utilize gloves with three color markers for gesture recognition, successfully detecting all twenty-six American Sign Language (ASL) alphabet letters. Ishiyama et al. [32] recognized 96

hand gestures by adding AR markers and structured markers to monochromatic gloves.

The other method is to utilize intrinsic features of the hand itself for recognition, such as skin color and edges [50], geometric features like length and width [75], and features of palm prints [84]. For example, Liu et al. [44] take geometric features of translation, rotation, and scale-invariant to finish gesture recognition. Nguyen et al. [53] identify hands in a gloved state using the hand's shape. Saboo et al. [61] use a two-level combination of color information, including skin filtering and motion information through three-frame differencing to recognize gestures. Singha et al. [65] also combine three-frame differencing and skin filtering to accomplish dynamic hand gesture recognition. Avola et al. [3] utilize a keypoint-based end-to-end framework for 3D hand tracking and pose estimation and apply it to the task of hand gesture recognition successfully. Additionally, some researchers use optical flow to capture the motion cues in areas including Time-to-Contact calculation [5], interaction between two people [8], and objects held by the robot contacting the environment [39], which proves the performance of optical flow.

Generally speaking, the former method requires users to wear glove-like devices to provide more accurate results but can be inconvenient for users, limiting naturalness [50]. In contrast, the latter method is more natural and does not require additional devices. In our work, we utilize the information generated by the motion and deformation of fingers and palms. We employ a frame-by-frame optical flow approach to characterize the deformation of palm prints after contact, concurrently utilizing information representing the motion trajectories of fingers over short periods. Moreover, we focus on detecting subtle touch states using global dense optical flow directly, while former studies focus on the obvious motion with local landmark optical flow.

3 Palmpad

We aim to utilize Palmpad to address the shortcomings of existing hand tracking on MR devices, particularly in-depth estimation, in the context of index-to-palm tasks. This will result in more precise and low-latency touch state detection, ultimately enhancing the user experience in index-to-palm interaction. To achieve this, in this section, we first analyze the problem, outlining the capabilities that Palmpad should provide and how these capabilities can be achieved. Subsequently, we present our approach to implementing Palmpad in terms of hardware and algorithms.

3.1 Consideration of Interaction Methods

In order for Palmpad to accurately encompass all interaction modes within index-to-palm interaction and to facilitate rapid deployment on the vast majority of current commercial MR devices, we need to consider the following research questions:

DQ1: What interaction modes should Palmpad provide? While existing work on index-to-palm interaction encompasses a variety of input methods, such as gestures [72], keyboard [71], and touchpad [22]. When we consolidate these methods, the primary touch tasks in index-to-palm interaction are index-click-palm and index-swipe-palm. Although both tasks fundamentally rely on precise touch state detection, the cognitive aspects and behaviors of users differ when performing these two types of actions. During clicking, users typically exhibit touch durations in the range of tens of milliseconds [15], with minimal interruptions, resulting in more significant differences between frames and blurrier fingertip images. In contrast, swiping involves longer, less predictable touch durations, potential pauses, and substantial relative displacement between the right index finger and the left palm. As a result, although both clicking and swiping tasks involve touch state recognition, they should be treated differently in the algorithm to achieve better recognition accuracy.

DQ2: What type of camera is suitable for Palmpad's task? Previous research has extensively explored solutions for touch detection using depth cameras or multiple cameras [22, 76, 77], which often estimate fingertip depth to determine touch states. They typically achieve spatial resolutions of around 1 centimeter. However, not all MR devices are equipped with depth cameras or multiple cameras. Although the power consumption of 3D tracking is improving, multi-camera systems occupy a larger physical space, and the computational power requirements and energy consumption for cameras themselves are several times greater than those of monocular cameras. To make Palmpad compatible with the majority of MR devices, we have opted for recognition using a single RGB camera. To capture full images of both hands during natural interaction, the camera's field of view (FOV) should be relatively wide. However, an excessively wide FOV can lead to image distortion, and some MR devices also use the camera for video recording purposes, so the FOV cannot be too large. Considering these factors, we selected a camera with a 120° FOV. To detect momentary touch events, such as clicks, the camera should have a high frame rate to capture more information. Therefore, we chose a camera with a frame rate of 120 frames per second (fps) to ensure abundant information.

(a) Images of index finger with clear outlines and shadows.

(b) Images of index finger with blur outlines and shadows.

Figure 2: Images of the index finger interacting on the palm.

DQ3: What visual features should be selected for Palmpad's task? Once camera parameters are determined, the next consideration is what features can be used to recognize touch state efficiently and accurately based on the existing hardware. Previous work often analyzed static images [1, 62], which is more suitable for

tasks like index-swipe-palm. In swiping tasks, the user's index finger moves at a moderate speed, without abrupt directional changes, resulting in smoother and clearer fingertip outlines and shadows (as shown in Fig. 2(a).). The features in single frames are relatively easy to extract. However, during clicking, the entire process is extremely rapid, and even with a 120fps camera, only a few frames can be recorded. Additionally, due to the high-speed motion and abrupt changes in direction when the finger contacts the palm and rebounds, the pixels around the finger appear very blurry in the images (as shown in Fig. 2(b).). Relying solely on single frames for recognition is not suitable in such scenarios, and introducing more temporal information is necessary. In this paper, we considered using both raw images and global optical flow computed between frames as inputs. Experimental evaluation showed that incorporating temporal information significantly enhances system accuracy and stability. Moreover, this approach outperformed solutions utilizing LSTM, highlighting the substantial advantage of our method in extracting temporal information.

3.2 Hardware Prototype

Figure 3: The hardware prototype and interaction method of Palmpad. A 120fps camera was fixed on the Quest 2 for input with an MR scene displayed through the Quest 2.

Based on the previous analysis, we select a camera with a frame rate of 120 fps, a 120° (diagonal) field of view (FOV), horizontal and vertical FOVs of 113° and 81°, and a resolution of 1280 × 720. We choose the Quest 2 as the MR head-mounted display (HMD) for our experiments. The camera is fixed above the Quest's center position using a rotatable bracket, and it is oriented diagonally downward at an angle of approximately 30° with respect to the vertical plane (as shown in Fig. 3). According to the pilot study, this setup ensures that both hands are fully captured when users engage in natural index-to-palm interaction.

3.3 Algorithms

3.3.1 Pipeline. According to the analysis in Section 3.1, we have designed the algorithm as shown in Figure 4. The system generally takes a sequence of n images with a time interval of dt as input and outputs the touch state (true or false) at the time corresponding to the last frame image. The system is implemented as a neural network (NN).

Before inputting, we crop the original frames and apply data augmentation of uniform parameter perturbations to enhance the system's robustness. Initially, we treat each frame as an independent input, aiming to extract static features related to the touch state. Subsequently, we consider using multiple frames in temporal sequence as input to extract temporal information. Finally, we integrate all this information to make a judgment about the touch state.

During training, we train our model for 100 epochs, with the initial learning rate set to 0.001, exponentially decreasing by a factor of 0.1 at epochs 15 and 50.

3.3.2 Image Processing. Due to computational constraints, directly inputting the original images(1280×720) into the neural network is not feasible. Therefore, image compression is required to preserve as much useful information as possible in the compressed images. As the changes in image features during index-to-palm interaction are primarily confined within the left hand's palm area, with other background regions remaining nearly static, and the variations on the palm are mainly concentrated around the tip of the right index finger, we utilize the Mediapipe hand landmark model [85]to determine the positions of the palm and fingers.

The Mediapipe hand landmark model can extract 21 key points for each hand from the image. We select the smallest rectangle encompassing all points as a cropping box to define the palm area and extend it by 10 pixels outward to ensure the entire palm is included within the image. To maintain a consistent aspect ratio, we adjust the cropping box to a square shape. Simultaneously, we crop a 128×128 square centered around the tip of the right index finger as the fingertip image input. Our real-time experiments have demonstrated that even when there are slight deviations between the key points provided by Mediapipe and the actual positions, this approach ensures that the tip of the right index finger remains within the image frame.

Due to the possibility that experimental data may not cover all usage scenarios, we have considered the following two data augmentation methods: 1) Random angle rotation. Although in actual use, the orientations of the camera and HMD are generally aligned, pilot studies have shown that the orientations of the left and right fingertips during natural index-to-palm interaction are not fixed, with an overall angle variation of around 30°. Therefore, we randomly rotate the input images by -30° to 30° to cover most angles during usage. 2) Random brightness jitter. Considering that the ambient brightness during actual use may vary, we proportionally adjust the brightness and contrast of the input images, with scaling factors uniformly distributed between 0.5 and 1.5.

3.3.3 Feature Extraction and Touch State Detection. Using the cropped and augmented images as input, we extract features through a fine-tuned CNN model with a pre-trained ResNet [26] as the backbone. The overall image of the left hand's palm and the image of the right index fingertip are input to their respective ResNet models, producing feature vectors of length 1000. This step yields the static features for the current input frame.

According to the analysis in Section 3.1, temporal information is crucial for touch state detection in Palmpad. Therefore, in the

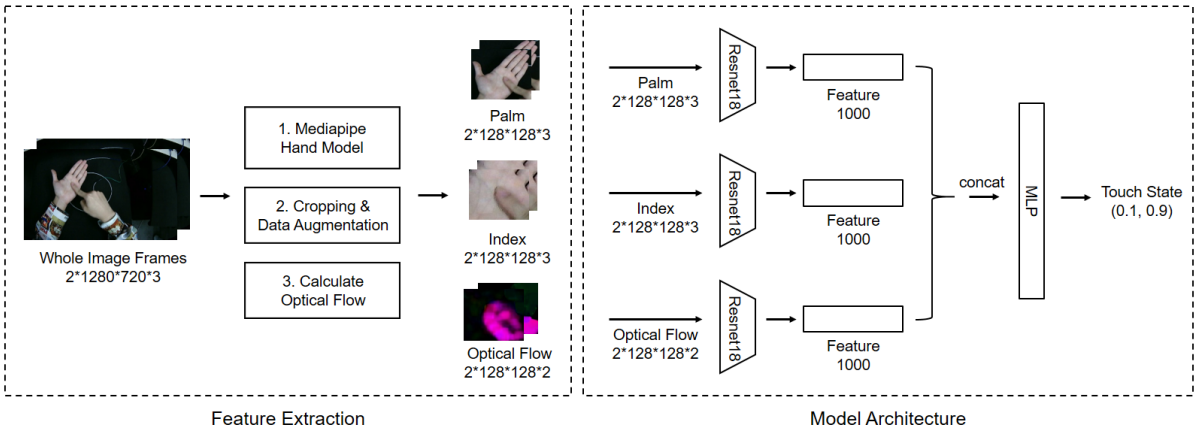


Figure 4: The algorithm pipeline of Palmpad. The original multi-frame images undergo key region extraction using Mediapipe (left palm and right index fingertip), followed by cropping and data augmentation. Then optical flow frames are calculated. Subsequently, raw frames, along with the optical flow motion information between adjacent frames, are fed into a ResNet-based CNN model for touch state classification.

algorithm, we need to consider how to effectively utilize this information. We note that using optical flow [12] can efficiently and explicitly extract motion information from consecutive temporal images. The assumption in optical flow that the overall image does not undergo particularly drastic changes holds true in the usage of Palmpad. This is because, whether it is index-click-palm or index-swipe-palm, the time interval between two frames is very short, and the main image changes are concentrated around the fingertip, making them easily trackable by optical flow. Therefore, we calculate the global dense optical flow to obtain the angular and displacement information for each pixel between consecutive fingertip frames using the bounding box cropped by Mediapipe. Since this information is still presented in two dimensions, we use a CNN with ResNet as the backbone to extract features from it. Finally, all the feature vectors to be classified, including static and temporal features, are concatenated and input into a fully connected network (using ReLU as the activation function and setting dropout to 0.5) for classification, yielding the touch state result for the current frame. To improve the robustness of optical flow, the dataset we collected contains head motion noise, including yawing, pitching, and random shaking, which makes the neural network more robust in the training. Moreover, in order to make the optical flow more stable, we utilize the landmarks generated by the Mediapipe to create a stable bounding box to focus on the hands and fingers to isolate the head movement.

Additionally, following the approach in similar works [41], we constructed a baseline model. This model, based on using CNN to extract features, takes the features and inputs them into an LSTM model [27] in temporal order, resulting in a feature vector containing temporal information. Subsequently, through comparative experiments, we will demonstrate the advantages of our model architecture in extracting temporal information for index-to-palm interactions.

3.3.4 Implementation. To balance accuracy and computational burden, we empirically set the input time length n to 2. The input time intervals are set to 1/120s, 1/60s, 1/30s, and 1/20s to accommodate different camera frame rates. We implemented the entire training and inference framework using Python and PyTorch on a Windows PC (CPU: Intel Core i9-9900KF; GPU: Nvidia GeForce RTX 2080Ti). Due to variations in the speeds of reading camera images, running the Mediapipe model for hand landmark inference, and conducting Palmpad model touch state inference, the overall framework is designed asynchronously using Python multiprocessing. The camera is connected to the PC via USB and is read in a separate process at a rate of 120fps. The Mediapipe model occupies a separate process, obtaining the current frame's image and updating hand landmarks at approximately 20fps. The Palmpad model occupies one process, taking image windows of size 2 as input, with a rate of around 100fps. We transmit the computed touch state results to the MR device (in this paper, we use Meta Quest 2 as the display HMD) via a socket connection.

4 ALGORITHM EVALUATION

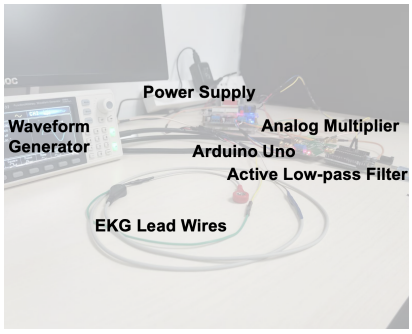
We systematically evaluated the sensing modality and algorithm we constructed to understand its specific performance under various settings and different parameters.

4.1 Participants

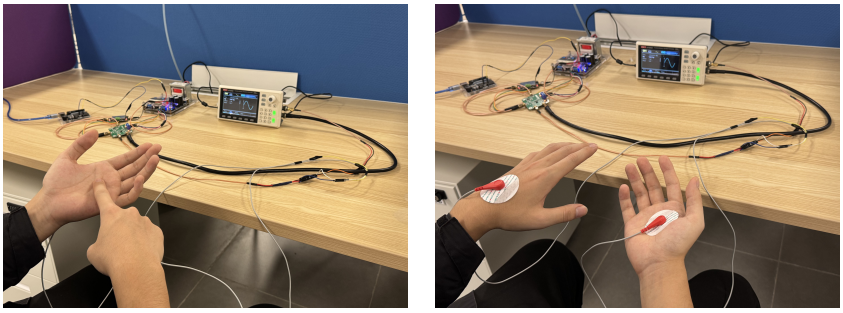
We recruited 16 participants (4 females) from the local campus by word-of-mouth to create our dataset, with an average age of 24.1(SD=2.5) ranging from 21 to 29 years old. All participants were right-handed.

4.2 Apparatus

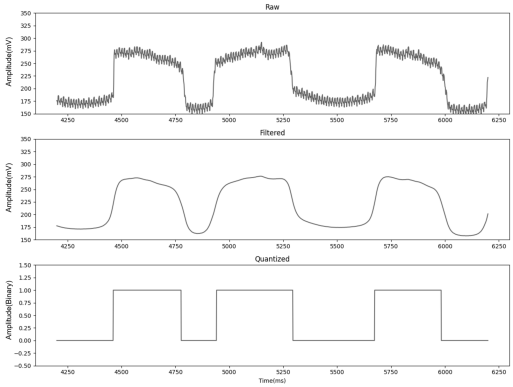
We expected to collect naturally comprehensive video data and automatically label the data. Therefore, during the data collection process, participants wore the video-capturing camera at the center of their forehead, ensuring that the camera consistently captured the entire hand region in its entirety. Participants would be requested to choose a natural posture to simulate the typical pose



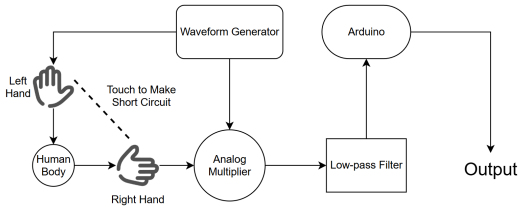
(a) An automatic touch state annotation system using alternating current signals.



(b) User connected to alternating signal circuit via adhesive electrodes.



(c) The electrical signals output by the hardware (Arduino Uno) and the results after filtering and quantization.



(d) A schematic diagram of an interactive circuit system including the human body, illustrating how RF signals are used to detect touch and processed by Arduino.

Figure 5: Illustrations of an automatic touch state annotation system using alternating current signals.

during normal use before collecting each video. Additionally, inspired by DualRing [42], we designed an automated label system. Participants wore electrodes of the automated label system to generate radio frequency signals indicating whether touch occurred. The overall setup of the data collection apparatus is depicted in Figure 5(a). How the user is connected to the circuit is shown as 5(b). We utilize a high-frequency alternating current circuit similar to DualRing [42] (as shown in 5(d)) to measure the impedance between the right index finger and the left palm, which allows us to detect the touch and the click, details of which can be found in Appendix C.

In the implementation of the circuit, we used the UTG932 waveform generator from UNI-T to produce two high-frequency sine wave signals at 12.5MHz . One signal remained constant, while the other signal passed through medical electrodes connected to the surfaces of hands, fixed on the back of the left hand and the palm of the right hand (to prevent interference from the electrodes when the camera captured data). Subsequently, we used the AD835 4-quadrant multiplier to multiply the two signals and amplify the result by 10 times. The output was then fed into a 1kHz low-pass filter to extract the low-frequency component. Finally, we used an Arduino Uno to sample the filtered signal at 1kHz and connected it to a PC via USB serial to obtain the digital signal. The sampled digital signal looks like Figure 5(c).

In the implementation of the software, we applied a one-Euro filter to smooth the signal . Subsequently, we utilized peak detection and peak width detection algorithms provided by the Scipy open-source library to quantize the smoothed signal, obtaining a binary signal (0 or 1) with the same frequency as the original signal. Following that, we aligned the binary signal with the video signal based on timestamps and downsampled the binary signal to obtain specific labels for each frame.

4.3 Data Collection

Before data collection, participants are briefly introduced to the basic principles of Palmpad and its potential applications. They will sign an informed consent form regarding the safe of high-frequency electrical signals passing through the human body during the experiment. Subsequently, with the assistance of the experimenter, participants wear the camera and electrodes, ensuring the correct configuration of the camera and the AC circuit. At the same time, the experimenter records the participants' age and information about the hand they are proficient in using.

The collection is divided into two parts respectively, collecting data on participants' clicks and swipes.

In the first part, participants will be asked to record videos of 36 clicks. They are required to use the right index to click on the left hand in a natural manner. Participants will be randomly instructed to perform the following actions in a specified order: clicking on the palm / finger, clicking with normal force / lightly / pretending to click, clicking rapidly (with intervals less than 0.5s) / sequentially (with intervals between 1-2s), clicking in order / randomly. They will click the same number of times (20 times for the palm and 12 times for the finger).

In the second part, participants will be asked to record videos of 26 swipes. They need to use the right index to slide on the left hand in a natural manner. Participants will be randomly instructed to perform the following actions in a specified order: sliding on palm / finger, sliding rapidly / slowly, sliding maintaining contact / pretending to slide / lift-slide-lift (only for palm operations), sliding horizontal / vertical(only for palm operations) / arbitrary direction.

After recording each video, participants adjust their body posture and camera placement according to their comfort, aiming to cover various camera perspectives in a natural state. Additionally, participants can choose to take a 30-second break between videos, and there is also a 30-second break between intervals in each of the three sections. All of the participants are divided into four groups and finish the data collection under different lighting conditions, which are morning light / midday light / evening light / artificial light. Each participant completes the data collection in approximately 45-60 minutes.

In the end, we collected a total of 16 participants × (36 segments in the first part + 26 segments in the second part) = 992 video segments. The average duration of each video segment is approximately 12.81 seconds, resulting in a total video length of 211 minutes. After post-processing, including filtering, quantization, alignment with the corresponding video, and downsampling, we generated annotations for each frame of each video segment, indicating touch or no touch. All the data we collected can be found at https://huggingface.co/datasets/Teburile/Palmpad_Dataset.

4.4 Dataset

To construct the dataset for training and evaluation, we sampled frame data from the collected 992 videos mentioned earlier. For videos containing both touch and non-touch instances (corresponding to normal force clicks and light taps in the tapping section and lift-slide-lift in the sliding section), we treated them as alternating positives and negatives. From each positive and negative instance, we selected one sample (e.g., 20 normal force clicks on the palm would yield 21 negative samples and 20 positive samples). For videos with only touch instances (corresponding to maintaining contact sliding in the sliding section) and videos with only non-touch instances (corresponding to pretending to tap in the tapping section, pretending to slide in the sliding section), we randomly sampled 40 samples.

For each sample, we selected a window of size 2 to select frames. Examples of the first frame of the samples can be seen in Figure 13. The label for the sample was determined to be the label of the last frame. In addition to the original 120fps dataset, we used a sampling method to generate 30fps and 60fps versions of each video, applying the same method to create corresponding datasets. We generated a dataset for each user's 36+26=62 videos, comprising approximately 900 positive samples and about 1450 negative samples.

We employed a cross-user training strategy, using three different time intervals (1/120 second, 1/60 second, 1/30 second, and 1/20 second). We employed Palmpad (utilizing optical flow to extract temporal information), Model-LSTM (utilizing LSTM to extract temporal information, the same architecture as [41]), model without optical flow (using multi-frame data with positional embedding), and a model without Mediapipe cropping (using only the full-size consecutive images of as input without cropping). We report the accuracy, recall, precision, and F1 score for touch state recognition.

4.5 Results

4.5.1 Model Performance. We conducted a leave-one-out evaluation, where each time, we selected the data of one participant as the test set, and the results (including F1 score, accuracy, recall, and precision) for all participants were summarized in Table 1. As shown in the results, Palmpad achieved the highest F1 score of 96.2% (SD=2.6%) and the highest accuracy of 97.0% (SD=2.0%) with an interval time of 1/30s when using the Palmpad model. Although the accuracy varied with different time intervals, the Friedman test results of F1 score($\chi^2(2) = 1.065, p = 0.786$) indicated no significant difference in accuracy among different time intervals. This means that the impact of different time intervals on our model is insignificant, and it can be effectively deployed on lower frame rate cameras, such as 20fps or maybe even lower. However, from the results, 30fps is currently the best in terms of performance. This frame rate is achievable by the majority of commercial cameras available today and can, therefore, be widely applied to various types of devices.

In addition, we randomly selected a 10-second segment of data and plotted it along the time axis. As shown in Figure 6, the moments identified by the model for press and lift are very close to the ground truth on the time axis (within 40ms). This indicates that our system has excellent real-time performance, as it can provide instantaneous results even during fast-tapping actions.

4.5.2 Comparison with other models. To verify the efficiency of Palmpad in extracting temporal information, we implemented a baseline that uses LSTM. Its main framework is similar to Palmpad, using the same CNN framework to extract features from each frame. The features are then concatenated in chronological order and inputted into LSTM. The output is concatenated with each frame feature and classified through a fully connected network. Compared to this LSTM-Model , our model significantly outperformed this baseline with each interval time . This not only validates our analysis that temporal information is crucial for touch state detection but also indicates that our model effectively extracts temporal information from continuous image inputs and is particularly well-suited for the index-to-palm interaction tasks.

Also, we reproduced the model mentioned in [62], which achieves the best performance that we can find. As they did not release the dataset, we reproduced their method on ours. On our dataset, this model achieves only an accuracy of 89.1% and an F1 score of 85.3%,

Table 1: F-1 scores, accuracies, recalls, and precisions of Palmpad evaluation with four types of models and four types of interval time. The numbers in the table are in percentage(%).

Interval Time	Palmpad				Model-LSTM				w/o Optical Flow				w/o Mediapipe			
	F1	Acc	Rec	Prec	F1	Acc	Rec	Prec	F1	Acc	Rec	Prec	F1	Acc	Rec	Prec
1/120s	96.1(2.3)	**97.0(1.9)**	**96.1(2.3)**	96.1(3.2)	94.2(3.2)	95.5(2.6)	94.4(3.7)	94.2(4.3)	94.5(3.0)	95.8(2.3)	94.6(4.0)	94.5(3.9)	94.1(2.9)	95.5(2.3)	94.4(3.7)	94.0(3.7)
1/60s	95.9(2.5)	96.8(2.0)	95.9(3.5)	96.0(3.3)	94.3(2.8)	95.7(2.1)	94.3(3.4)	94.5(4.0)	94.5(2.9)	95.8(2.2)	94.4(3.7)	94.6(3.0)	94.2(3.1)	95.5(2.4)	94.3(3.6)	94.2(3.9)
1/30s	**96.2(2.6)**	**97.0(2.0)**	**96.1(3.1)**	**96.3(2.7)**	94.2(2.5)	95.6(1.9)	94.0(4.0)	94.6(3.3)	94.9(2.5)	96.1(1.9)	94.4(3.7)	95.5(2.9)	94.5(2.6)	95.8(2.0)	94.3(3.1)	94.8(3.6)
1/20s	95.9(2.2)	96.8(1.8)	95.9(3.0)	95.9(3.5)	93.4(3.1)	94.9(2.5)	93.2(4.5)	93.8(4.7)	94.5(2.8)	95.8(2.1)	94.3(3.8)	94.7(3.2)	94.0(3.3)	95.4(2.4)	94.3(5.1)	94.0(3.9)

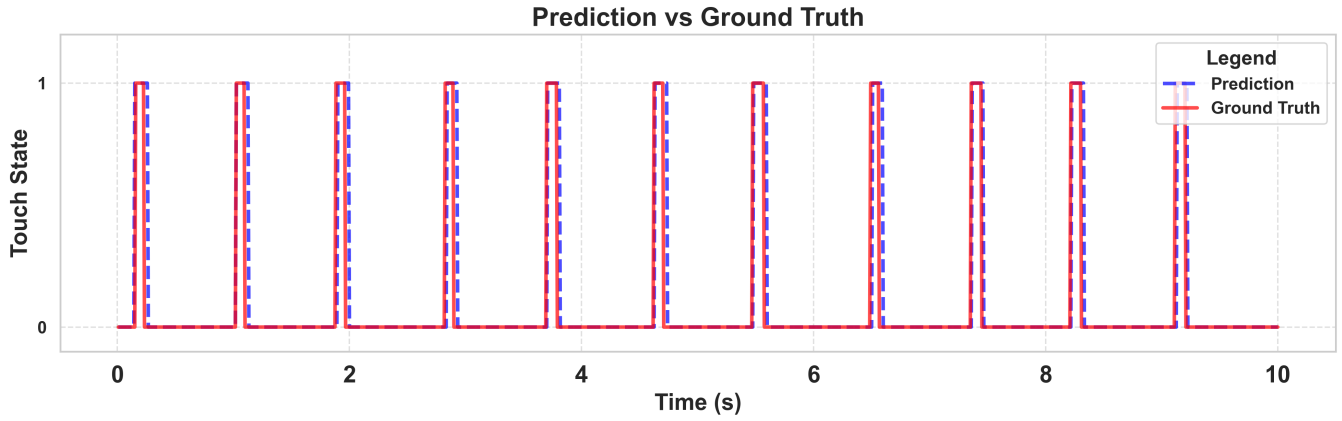
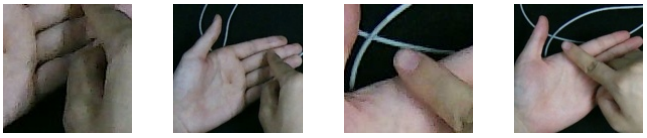


Figure 6: A 10-second data segment from real usage, including ground truth and model prediction results. The time difference between all rising and falling edges predicted by the model and the true values does not exceed 40ms.

which is much lower than the performance of our model and significantly deviates from the results reported in their paper. We believe this is because our dataset is relatively more complex, with factors such as different tap speeds, different tap positions, and camera shake during natural wearing, all of which may cause this model to perform poorly on our dataset. Additionally, we retrained PressureVision++ [13] on our dataset and got an accuracy of 95.8% and an F1 score of 94.5%. In a click event, the palm deformation mitigates the expression of the pressure, leading to incorrect estimation.

4.5.3 Ablation Study. In Section 3.1, we pointed out that temporal features and Mediapipe-based cropping play significant roles in touch state detection. Here, we trained a model with the same architecture but with positional embeddings (the same as Transformer[69]) instead of calculating optical flow , and another model without using Mediapipe for cropping. The evaluation results show that they are lower than Palmpad in terms of accuracy indicators (as shown in Table 1).

4.5.4 Bad Cases. We manually examined 100 error cases, including 50 false negatives and 50 false positives. Among them, 25% of the errors were attributed to inaccuracies in Mediapipe hand tracking (as shown in Figure 7(a)). Such errors could be improved by employing a more accurate hand-tracking system. In addition, we also discovered some errors caused by finger-click positions being near the boundary of the palm or close to other objects (as shown in Figure 7(c)). These issues can be addressed through additional algorithms or by avoiding such situations during usage.



(a) An error case caused by inaccuracies in the Mediapipe fingertip position estimation (fingertip).

(b) An error case caused by inaccuracies in the Mediapipe fingertip position estimation (whole palm).

(c) An error case caused by finger-click positions being near the boundary of the palm or close to other objects (fingertip).

(d) An error case caused by finger-click positions being near the boundary of the palm or close to other objects (whole palm).

Figure 7: Examples of the error cases.

5 USABILITY EVALUATION AND APPLICATIONS

5.1 System Implementation

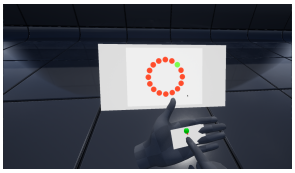
Finally, we implemented a real-time system following the approach outlined in Section 3.3.4 to evaluate the interactive experience of Palmpad in an actual MR environment. In this system, we transmitted Palmpad's touch state to the MR device through a socket, enabling basic functions like swipe and click. The user's finger position information was provided by the built-in hand tracking of the Quest 2.

Additionally, we implemented two comparative systems. The first system referred to as the Threshold system, utilized the hand

tracking of the Quest 2 to calculate the distance between the right fingertip coordinates and the left palm plane. If this distance fell below a certain threshold, the touch state was set to true; otherwise, it was set to false. The second system, known as the Collision system, employed the Quest 2's hand tracking to generate meshes for both hands. The touch state was set to true when the meshes collided and false otherwise.

Due to difficulties faced by users in the pilot experiment with the Collision, mainly due to its inaccurate depth estimation leading to an unstable touch state, we focused on comparing Palmpad with the Threshold system in the final experiment. Two interaction tasks were designed, and corresponding interaction scenarios were set up for evaluation.

5.2 Study Design



(a) MR interface display for the Fitts's Law experiment. The random green circle represents the next target. Participants need to move the cursor to the target position using the touchpad supported by Palmpad or Threshold method and lift a finger.

(b) MR interface display for the clicking experiment. Participants need to consecutively click on the position of the red square and report the number of errors based on the feedback provided by the system.

Figure 8: The tasks of the usability evaluation experiment.

To assess the system's usability in index-swipe-palm interaction, we designed a Fitts's Law experiment [11]. In a virtual environment, we placed a display screen with 16 target circles uniformly distributed around a large circular perimeter(as shown in Fig. 8(a).). For each task, a random circle was selected as the starting point, the cursor was set to the center of the circle, and another circle was chosen as the endpoint, marked in green. During each task, the user had to move the cursor from the starting point to the endpoint as quickly as possible and lift it. If, at the moment of lift, the cursor was within the circular area, the task was considered successful; otherwise, the user had to retry. In the experiment, we categorized the sizes of the circles into large, medium, and small. Through pilot experiments, we determined that the smallest size ensured users could complete the task after a few attempts without being too small to stably control the cursor on it. In addition to controlling the circle sizes, we managed the distance traveled in each task. In a circle encompassing 16 target circles, there were 8 different distances between any two circles. We randomly selected one distance for each task, conducting 2 trials for each distance. Therefore, in this experiment, each system performed a total of 8 (different movement distances) × 2 (trials) × 3 (circle sizes) = 48 swipes. We fine-tuned the maximum distance to ensure it could be completed in a single swipe. Therefore, if multiple swipes occurred within one task, it was due to the user having difficulty controlling the cursor to stay within the circular area at the moment of lift, resulting in multiple attempts. The number of swipes within each task in this experiment served as a measure of the system's ease of use.

Secondly, to assess the system's usability in index-click-palm interaction, we designed an experiment to measure click accuracy. In this experiment, users were instructed to perform four consecutive clicks on their palms at a natural frequency. If successful, users could observe the counter's number change after each click. After each set of four consecutive clicks, users had to report the total number of misfires and misses. Each system underwent 20 blocks, totaling 80 clicks.

In the experiment, half of the users started with Palmpad, while the other half began with the Threshold system to complete the tasks. Similarly, half of the users started with the index-swipe-palm task, while the other half began with the index-click-palm task. During the experiment, users could choose to take breaks of any length between tasks to ensure their engagement in each set of experiments. After each session, users were asked to fill out the NASA-TLX scale, rate some questions derived from the SUS scale, and provide subjective evaluations if they were willing.

5.3 Participants and Apparatus

In the experiment, we randomly invited 12 participants from the university campus (4 females, 8 males, Age=23.00±2.95) who had not participated in the experiments of Study 1. We collected information about their experience with VR hand tracking, with scores of 3.00±2.04. A score of 0 indicates that they have never used or heard of it before, and a score of 6 represents proficient usage with development experience. The participants were all right-handed, as required by the experimental setup.

5.4 Results

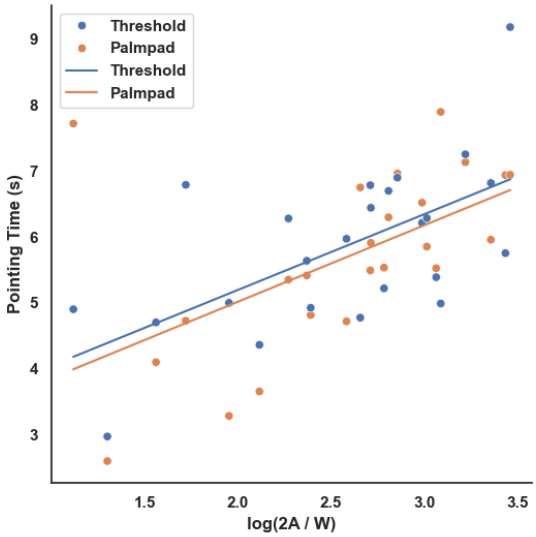


Figure 9: The result of the Fitts's Law evaluation experiment.

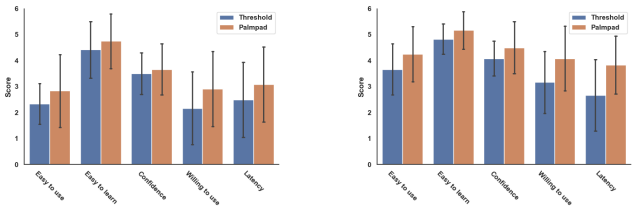
5.4.1 Performance Comparison. We compared the Palmpad solution with the Threshold solution in the Fitts's Law experiment and the clicking experiment.

In the Fitts's Law experiment, we recorded the time and distance of each of the 48 swipes for each user, as well as the radius of the target to be reached for each operation. The overall average sliding time for Palmpad was 5.65 seconds (SD=4.38), with an average number of swipes of 1.65 times (SD=0.99). The overall average sliding time for Threshold was 5.84 seconds (SD=4.54), with an average number of swipes of 1.70 times (SD=1.00). Palmpad reduced the task completion time by 3.1%. In addition, we fitted the Fitts's Law curve based on the movement distance and target size, as shown in Figure 9. From this, we derived an average throughput of 21.40 for Palmpad and 20.98 for Threshold. Results showed that Palmpad slightly outperformed Threshold, the reason for which may be that the task of swiping needed continuous contact between finger and palm rather than frequent touch down and up, where Palmpad and Threshold both had stable performance. For Thresholds based on hand tracking, it is common to keep a low or even minus(model clipping) value, which ensures the stability of its performance.

In the clicking experiment, we recorded the number of false touches and misses reported by each user for each click and calculated precision and recall. In a total of 960 click experiments among all users, both Palmpad and Threshold solutions had very few false touch occurrences (less than 15). However, there was a difference in the number of misses. Palmpad achieves an accuracy of 95.3% while the Threshold is only 92.5%, whereas the Palmpad improves the accuracy by 2.8%. We used the Wilcoxon signed-rank test to examine the significance of the difference. In terms of precision, there was no significant difference between the two solutions ($z = -2.86, p = 0.461$). In terms of recall, Palmpad significantly outperformed the Threshold solution ($z = -2.27, p = 0.045$). This indicates that the Palmpad solution is particularly well-suited for scenarios that require rapid clicking.

Additionally, we measured the sensing resolution of each scheme to distinguish their capability to discern distances between fingers and the palm. For the Palmpad scheme, under normal posture, the minimum distinguishable distance that can be steadily reported during the touch-down to touch-up process is 10mm (measured by a millimeter scale at the same depth of the touch point from the recording of an along-surface macro camera), and during the touch up to touch down process, this distance is 5mm. For the threshold-based scheme, the distances are 20mm and 10mm, respectively. Therefore, Palmpad's spatial resolution has improved by 50%, and such measurements confirm that Palmpad can achieve a significantly higher resolution in distinguishing subtle near-surface touch states.

5.4.2 Questionnaire Results. To evaluate users' experiences and subjective ratings of different solutions, we designed a questionnaire based on the NASA-TLX scale and the SUS (System Usability Scale), which includes eleven questions covering aspects of mental demand, physical demand, performance, effort, temporal demand, frustration, easiness to use, learn-ability, willingness to use, confidence and latency when using the Palmpad and Threshold schemes. Figure 10 shows part of the results from the SUS scale. Higher scores indicate that the system is easier to use and easier to learn, gives higher confidence, higher willingness to use, and lower latency. The Appendix Figure 12 shows the results from the NASA-TLX scale. Lower scores indicate lower demand, better performance, and less frustration when using the system. The ratings indicate that participants generally considered the Palmpad scheme to be better than the Threshold scheme. The former obtained higher scores in ease of use, confidence, and willingness to use. It also had lower demand than the latter.



(a) Scores for the Fitts's Law task. (b) Scores for the clicking task.

Figure 10: Subjective rating scores for different usability evaluation tasks. 0 - strongly disagree, 6 - strongly agree.(SUS)

5.4.3 Subjective Feedback. In addition to the quantitative ratings from the scales, we also collected some subjective feedback from users according to their willingness. They showed a positive attitude towards the potential of Palmpad. *"The Threshold method is not as sensitive as Palmpad."*(P3) *"The Palmpad method is more user-friendly; its accuracy is higher than the Threshold method; I am unwilling to use the Threshold method because it has a greater delay and wastes time."*(P4) *"Palmpad is smoother when lifting hands, making it very suitable for keyboards."*(P8) Some also believe that Palmpad is inferior to Threshold because *"it occasionally registers false touches in the air."*(P5) Additionally, some participants expressed concerns about the task design, *"The screen during the sliding operation is fixed above the left hand, which requires me to constantly turn my head to look at it, putting a significant strain on my neck."*(P8) The score for the sliding task is slightly lower than that for the clicking task, which may be due to the mouse's position mapping during sliding relying on somewhat shaky hand tracking, resulting in some jitter in mouse control. *"This mouse is a bit shaky when sliding, I would be more willing to use it if it could be better optimized."*(P7)

5.5 More Application Scenarios

As mentioned in Section 3.1, the majority of index-to-palm applications can be supported by index-click-palm and index-swipe-palm, and Palmpad supports these two operations, thus enabling these applications. For example, we have implemented an always-available keyboard in MR (Mixed Reality) using the precise click functionality provided by Palmpad and position information provided by the built-in hand tracking (as shown in Fig.11). Based on this, we have provided a highly available virtual keyboard and mouse in the MR environment, theoretically capable of completing all 2D GUI interaction tasks. In addition, since we can project any information onto the palm and provide touch detection, the interface, and logic of smartphones can be easily transferred to MR devices, where we can use the palm as an always-carried smartphone in MR.

Figure 11: The keyboard application prototype supported by Palmpad in MR scene.

Additionally, although the technology mentioned is oriented towards MR scenarios, and all designs are crafted for interactions in MR, Palmpad is not limited to MR. Small wearable devices equipped with a single RGB camera are also endowed with touch detection capabilities, such as Ai pin [30]. Through Palmpad, these devices can also accurately determine the touch state, which provides more dimensions for their interaction space.

6 LIMITATION AND DISCUSSION

In this section, we discuss the limitations of Palmpad, which suggests several new directions for future work.

6.1 System Robustness

Although Palmpad has gained a good performance on the dataset since it adopts a vision-based sensing solution in the practical implementation, it encounters a few bad cases due to the unavoidable influence of the camera, such as extreme user postures, unsuitable ambient lighting, intentional obstructions and so on. On the one hand, the postures of the user or the obstructions may make the palm and finger invisible in the camera view. On the other hand, overly dark or overexposed environments can affect hand tracking. For instance, when the distal phalanx is vertical, it can cause self-occlusion of the fingertip image. Although such scenarios may slightly reduce Palmpad's performance, they are infrequent in typical interactions, especially given the top-down viewing angle provided by a head-mounted camera. The moving speed also matters. When the index finger clicks particularly fast, MediaPipe is unable to quickly and stably capture the boundaries of the palm and fingers.

In order to alleviate these issues and enhance the robustness of the entire system, there are several possible solutions, which include: (1) add additional camera angles to compensate for occlusions in some bad cases, (2) use active light sources to adjust for ambient lighting conditions, (3) add markers to improve hand tracking performance during rapid movements. All of these methods deserve further research to improve the performance of Palmpad in all kinds of cases.

Additionally, further research that pays attention to the improvement of speed and stability in obtaining hand boundaries is needed. From our empirical observation, many of the bad cases of the performance of the entire system are caused by the mistakes of hand boundaries. We believe that more precise and faster hand boundaries are of great practical value.

Moreover, We apologize for the inconvenience regarding skin color, manicure, tattoos, and other factors that may affect the model's performance. However, the purpose of this study is to validate feasibility. Subsequently, more extensive data from a broader population can be collected, and these issues can be easily addressed based on the same framework.

6.2 Form Factor and Implementation

The hardware prototype of Palmpad currently includes an external RGB camera placed on the upper surface of the MR headset, which results in an increase in the weight of the MR headset and a shift in the center of gravity. This implementation is designed to validate the feasibility of our system and maintain the convenience of debugging. Additionally, the main computation tasks are performed on the PC, which facilitates the evaluation and optimization of the entire system and guarantees a stable running rate for the pipeline. In the future, it is highly likely to use the MR headset's front-facing cameras instead of external RGB cameras and utilize its built-in hand tracking to obtain hand boundaries while simultaneously integrating our current model(35M) into the MR headset.

As mentioned in Table 1, the whole system gains the best performance at a resolution of 1280×720 and 30 fps, which means that the common front-facing RGB cameras in MR headsets(1080p and 30fps in HoloLens 2 [48]) meet this requirement. As for the computational cost and energy consumption, we implemented the prototype of our algorithms on a PC with strong computational performance, which keeps the pipeline running at a constant frame rate. Given the model size(35M) we used for our pipeline, it is lightweight enough to be implemented. Additionally, reusing the result of hand tracking from an MR headset provides the solution of cropping the input image with lower consumption than Mediapipe. We believe that Palmpad is capable of deployment on a commodity MR system.

6.3 Future Works

So far, our system only implements the interaction of the right index finger with the left palm. In future work, expanding the area where interaction is possible will be very valuable. In our future work, we plan to support multi-finger operations, as well as the potential for tapping and sliding on both the palm and back of the hand. We will also provide symmetric services for left-handed users. We believe that collecting sufficient data for different skin colors, palm texture features, tattoos, and nail art to conduct more detailed iterations and evaluations is of great practical value.

In addition, although the deformation of the palm does not affect the accuracy of the Palmpad touch state, it does have a certain impact on applications (such as cursor movement or typing). This is because currently, for convenience, we simply place the interface of the application above a certain distance from the surface of the palm. This may lead to the generation of 'parallax' artifacts and

thus affect user experience. Therefore, in the future, we need to design a more reasonable way to map interfaces such as keyboards onto palms. For example, by mapping recognized skeletal points on hands to certain fixed points on keyboards, the relative positions between keyboards and hands will remain consistent and the impact of palm deformation on user experience will be reduced.

7 CONCLUSION

In this work, we introduced Palmpad, a system characterized by high usability and ease of deployment, specifically designed for the detection of index-to-palm touch states in MR environments, utilizing only a single RGB camera. We conducted a detailed analysis of the features associated with index-to-palm interactions and devised an efficient CNN-based model for index-to-palm touch recognition, which incorporated varying levels of image features as input. We provided a prototypical implementation of Palmpad and our proposed model achieved an impressive touch recognition accuracy of 97.0%. Subsequent to a user study, Palmpad demonstrated superior performance compared to state-of-the-art commercial solutions, exhibiting a noteworthy 2.8% improvement in accuracy and a substantial 50% enhancement in spatial resolution. Lastly, the extensive potential of Palmpad for application was explored. Our usability evaluation study validated the enhanced usability of Palmpad over the threshold-based approach.

Acknowledgments

This work is supported by the National Key Research and Development Plan of China under Grant No. 2024YFB4505500 & 2024YFB4505502, Beijing Key Lab of Networked Multimedia, Institute for Artificial Intelligence, Tsinghua University (THUAI), Beijing National Research Center for Information Science and Technology (BNRist), 2025 Key Technological Innovation Program of Ningbo City under Grant No. 2022Z080, Beijing Municipal Science and Technology Commission, Administrative Commission of Zhongguancun Science Park No.Z221100006722018, and Science and Technology Innovation Key R&D Program of Chongqing.

References

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

[2] Daniel Ashbrook, James Clawson, Kent Lyons, Thad Starner, and Nirmal Patel. 2008. Quickdraw: the impact of mobility and on-body placement on device access time. *ACM* (2008). doi:10.1145/1357054.1357092

[3] Danilo Avola, Luigi Cinque, Alessio Fagioli, Gian Luca Foresti, Adriano Fragomeni, and Daniele Pannone. 2022. 3D hand pose and shape estimation from RGB images for keypoint-based hand gesture recognition. *Pattern Recognition* 129 (Sept. 2022), 108762. doi:10.1016/j.patcog.2022.108762

[4] Ananta Narayanan Balaji and Li-Shiuan Peh. 2021. AI-on-skin: Enabling On-body AI Inference for Wearable Artificial Skin Interfaces. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–7.

[5] TA Camus. 1995. Calculating time-to-contact using real-time quantized optical flow. (1995).

[6] Liwei Chan, Rong-Hao Liang, Ming-Chang Tsai, Kai-Yin Cheng, Chao-Huai Su, Mike Y Chen, Wen-Huang Cheng, and Bing-Yu Chen. 2013. FingerPad: private and subtle interaction using fingertips. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. 255–260.

[7] Qing Chen. 2008. Real-time vision-based hand tracking and gesture recognition. *null* (2008). doi:10.20381/ruor-12994

[8] Qingshuang Chen, He Li, Rana Abu-Zhaya, Amanda Seidl, Fengqing Zhu, and Edward J Delp. 2016. Touch event recognition for human interaction. *Electronic Imaging* 28 (2016), 1–6.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[10] Niloofar Dezfuli, Mohammadreza Khalilbeigi, Jochen Huber, Florian Müller, and Max Mühlhäuser. 2012. PalmRC: imaginary palm-based remote control for eyes-free television interaction. In *Proceedings of the 10th European conference on Interactive tv and video*. 27–34.

[11] Paul M Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology* 47, 6 (1954), 381.

[12] David Fleet and Yair Weiss. 2006. Optical flow estimation. In *Handbook of mathematical models in computer vision*. Springer, 237–257.

[13] Patrick Grady, Jeremy A Collins, Chengcheng Tang, Christopher D Twigg, Kunal Aneja, James Hays, and Charles C Kemp. 2024. PressureVision++: Estimating Fingertip Pressure from Diverse RGB Images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 8698–8708.

[14] 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 *European Conference on Computer Vision*. Springer, 328–345.

[15] 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*. 1059–1070.

[16] Hung-Jui Guo and Balakrishnan Prabhakaran. 2022. Hololens 2 technical evaluation as mixed reality guide. *arXiv preprint arXiv:2207.09554* (2022).

[17] Sean Gustafson, Christian Holz, and Patrick Baudisch. 2011. Imaginary phone: learning imaginary interfaces by transferring spatial memory from a familiar device. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. 283–292.

[18] Sean G Gustafson, Bernhard Rabe, and Patrick M Baudisch. 2013. Understanding palm-based imaginary interfaces: the role of visual and tactile cues when browsing. In *Proceedings of the SIGCHI Conference on human factors in computing systems*. 889–898.

[19] Shangchen Han, Beibei Liu, Randi Cabezas, Christopher D Twigg, Peizhao Zhang, Jeff Petkau, Tsz-Ho Yu, Chun-Jung Tai, Muzaffer Akbay, Zheng Wang, et al. 2020. MEgATrack: monochrome egocentric articulated hand-tracking for virtual reality. *ACM Transactions on Graphics (ToG)* 39, 4 (2020), 87–1.

[20] Shangchen Han, Beibei Liu, Robert Wang, Yuting Ye, Christopher D Twigg, and Kenrick Kin. 2018. Online optical marker-based hand tracking with deep labels. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–10.

[21] Ltd. Hangzhou Lingban Technology Co. 2023. rokid air pro. https://rokid.ai/rokid-air-pro/

[22] Chris Harrison, Hrvoje Benko, and Andrew D. Wilson. 2011. OmniTouch: wearable multitouch interaction everywhere. *ACM* (2011). doi:10.1145/2047196.2047255

[23] Chris Harrison and Haakon Faste. 2014. Implications of location and touch for on-body projected interfaces. *ACM* (2014). doi:10.1145/2598510.2598587

[24] Chris Harrison and Scott E. Hudson. 2008. Scratch input: creating large, inexpensive, unpowered and mobile finger input surfaces. *ACM* (2008). doi:10.1145/1449715.1449747

[25] Chris Harrison, Desney Tan, and Dan Morris. 2010. Skinput: appropriating the body as an input surface. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 453–462.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[27] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

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

[29] Da-Yuan Huang, Liwei Chan, Shuo Yang, Fan Wang, Rong-Hao Liang, De-Nian Yang, Yi-Ping Hung, and Bing-Yu Chen. 2016. Digitspace: Designing thumb-to-fingers touch interfaces for one-handed and eyes-free interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 1526–1537.

[30] Inc. Humane. 2023. ai pin. https://hu.ma.ne/aipin

[31] Hafiz Imtiaz and Shaikh Anowarul Fattah. 2013. A wavelet-based dominant feature extraction algorithm for palm-print recognition. *Digital Signal Processing* (2013). doi:10.1016/j.dsp.2012.06.016

[32] Hidetoshi Ishiyama and Shuichi Kurabayashi. 2016. Monochrome glove: A robust real-time hand gesture recognition method by using a fabric glove with design of structured markers. In *2016 IEEE Virtual Reality (VR)*. 187–188. doi:10.1109/VR.2016.7504716

[33] Hyoyoung Jeong, John A Rogers, and Shuai Xu. 2020. Continuous on-body sensing for the COVID-19 pandemic: Gaps and opportunities. *Science Advances* 6, 36 (2020), eabd4794.

[34] Hsin-Liu Kao, Artem Dementyev, Joseph A Paradiso, and Chris Schmandt. 2015. NailO: fingernails as an input surface. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 3015–3018.

[35] Naoaki Kashiwagi, Yuta Sugiura, Natsuki Miyata, Mitsunori Tada, Maki Sugimoto, and Hideo Saito. 2017. Measuring Grasp Posture Using an Embedded Camera. *IEEE* (2017). doi:10.1109/wacvw.2017.14

[36] M. Kasiselvanathan, V. Sangeetha, and A. Kalaiselvi. null. Palm pattern recognition using scale invariant feature transform. *International journal of intelligence and sustainable computing* (null). doi:10.1504/ijisc.2020.104826

[37] Wolf Kienzle, Eric Whitmire, Chris Rittaler, and Hrvoje Benko. 2021. Electroring: Subtle pinch and touch detection with a ring. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–12.

[38] Jin Hee (Heather) Kim, Kunpeng Huang, Simone L White, Melissa Conroy, and Cindy Hsin-Liu Kao. 2021. KnitDermis: Fabricating Tactile On-Body Interfaces Through Machine Knitting. *ACM* (2021). doi:10.1145/3461778.3462007

[39] Leon Kim, Yunshuang Li, Michael Posa, and Dinesh Jayaraman. 2023. Im2Contact: Vision-Based Contact Localization Without Touch or Force Sensing. In *Conference on Robot Learning*. PMLR, 1533–1546.

[40] Yuki Kubo, Yuto Koguchi, Buntarou Shizuki, Shin Takahashi, and Otmar Hilliges. 2019. AudioTouch: Minimally Invasive Sensing of Micro-Gestures via Active Bio-Acoustic Sensing. *ACM* (2019). doi:10.1145/3338286.3340147

[41] 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*. 1–14.

[42] Chen Liang, Chun Yu, Yue Qin, Yuntao Wang, and Yuanchun Shi. 2021. DualRing: Enabling Subtle and Expressive Hand Interaction with Dual IMU Rings. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* (2021). doi:10.1145/3478114

[43] Jhe-Wei Lin, Chiuan Wang, Yi Yao Huang, Kuan-Ting Chou, Hsuan-Yu Chen, Wei-Luan Tseng, and Mike Y Chen. 2015. Backhand: Sensing hand gestures via back of the hand. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 557–564.

[44] Liwei Liu, Junliang Xing, Haizhou Ai, and Xiang Ruan. 2012. Hand posture recognition using finger geometric feature. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. IEEE, 565–568.

[45] Yiqin Lu, Bingjian Huang, Chun Yu, Guahong Liu, and Yuanchun Shi. 2020. Designing and evaluating hand-to-hand gestures with dual commodity wrist-worn devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020), 1–27.

[46] Damien Masson, Alix Goguey, Sylvain Malacria, and Géry Casiez. 2017. WhichFingers: Identifying Fingers on Touch Surfaces and Keyboards using Vibration Sensors. *ACM* (2017). doi:10.1145/3126594.3126619

[47] Adnan Mehmood, Han He, Xiaochen Chen, Aleksi Vianto, Ville Vianto, Oğuz 'Oz' Buruk, and Johanna Virkki. 2020. Clothface: a passive RFID-based human-technology interface on a shirtsleeve. *Advances in Human-Computer Interaction* 2020 (2020), 1–8.

[48] Microsoft. 2024. HoloLens 2. https://www.microsoft.com/en-us/hololens/hardware#document-experiences

[49] Mohd Norzali Haji Mohd, Mohd Shahrimie Mohd Asaari, Ong Lay Ping, and Bakhtiar Affendi Rosdi. 2023. Vision-Based Hand Detection and Tracking Using Fusion of Kernelized Correlation Filter and Single-Shot Detection. *Applied Sciences* (2023). doi:10.3390/app13137433

[50] D. Mohr and G. Zachmann. 2013. A Survey of Vision-Based Markerless Hand Tracking Approaches. *null* (2013). doi:null

[51] 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

[52] Md. Haider ALI Md. HASANUZZAMAN Muhammad Aminur RAHAMAN, Mahmood JASIM. 2020. Bangla language modeling algorithm for automatic recognition of hand-sign-spelled Bangla sign language. *Frontiers of Computer Science* 14, 3, Article 143302 (2020), 0 pages. doi:10.1007/s11704-018-7253-3

[53] Binh P. Nguyen, Wei-Liang Tay, and Chee-Kong Chui. 2015. Robust Biometric Recognition From Palm Depth Images for Gloved Hands. *IEEE Transactions on Human-Machine Systems* (2015). doi:10.1109/thms.2015.2453203

[54] Uran Oh and Leah Findlater. 2014. Design of and subjective response to on-body input for people with visual impairments. *ACM Press* (2014). doi:10.1145/2661334.2661376

[55] Munir Oudah, Ali Al-Naji, and Javaan Chahl. 2020. Hand gesture recognition based on computer vision: a review of techniques. *journal of Imaging* 6, 8 (2020), 73.

[56] Joseph A. Paradiso and Che King Leo. 2005. Tracking and characterizing knocks atop large interactive displays. *Sensor Review* (2005). doi:10.1108/02602280510585727

[57] Huafeng Qin, Huafeng Qin, Jakob Puchinger, and Mounim A. El-Yacoubi. 2017. Deep Representation-Based Feature Extraction and Recovering for Finger-Vein Verification. *IEEE Transactions on Information Forensics and Security* (2017). doi:10.1109/tifs.2017.2689724

[58] Muhammad Rahaman, Mahmood Jasim, Md Ali, Tao Zhang, and M. Hasanuzzaman. 2018. A real-time hand-signs segmentation and classification system using fuzzy rule based RGB model and grid-pattern analysis. *Frontiers of Computer Science* 12 (11 2018). doi:10.1007/s11704-018-7082-4

[59] Ltd. Rayneo technology Co. 2023. rayneo-x2. https://rayneo.cn/product/x2/

[60] Mark Richardson, Fadi Botros, Yangyang Shi, Pinhao Guo, Bradford J Snow, Linguang Zhang, Jingming Dong, Keith Vertanen, Shugao Ma, and Robert Wang. 2024. StegoType: Surface Typing from Egocentric Cameras. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. 1–14.

[61] Shweta Saboo and Joyeeta Singha. 2021. Vision based two-level hand tracking system for dynamic hand gestures in indoor environment. *Multimedia Tools and Applications* (2021). doi:10.1007/s11042-021-10669-7

[62] Yuto Sekiya, Takeshi Umezawa, and Noritaka Osawa. 2021. Detection of Finger Contact with Skin Based on Shadows and Texture Around Fingertips. In *Human-Computer Interaction. Interaction Techniques and Novel Applications: Thematic Area, HCI 2021, Held as Part of the 23rd HCI International Conference, HCII 2021, Virtual Event, July 24–29, 2021, Proceedings, Part II 23*. Springer, 109–122.

[63] Yilei Shi, Haimo Zhang, Kaixing Zhao, Jiashuo Cao, Mengmeng Sun, and Suranga Nanayakkara. 2020. Ready, steady, touch! sensing physical contact with a finger-mounted IMU. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 2 (2020), 1–25.

[64] George K. Sidiropoulos, Polixeni Kiratsa, Polixeni Kiratsa, Petros Chatzipetrou, and George A. Papakostas. 2021. Feature Extraction for Finger-Vein-Based Identity Recognition. *Journal of Imaging* (2021). doi:10.3390/jimaging7050089

[65] Joyeeta Singha, Amarjit Roy, and Rabul Hussain Laskar. 2018. Dynamic hand gesture recognition using vision-based approach for human—computer interaction. *Neural Computing and Applications* (2018). doi:10.1007/s00521-016-2525-z

[66] Srinath Sridhar, Anders Markussen, Antti Oulasvirta, Christian Theobalt, and Sebastian Boring. 2017. Watchsense: On-and above-skin input sensing through a wearable depth sensor. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 3891–3902.

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

[68] 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

[69] A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).

[70] Velko Vechev, Alexandru Dancu, Simon T. Perrault, Quentin Roy, Morten Fjeld, and Shengdong Zhao. 2018. Movespace: on-body athletic interaction for running and cycling. *ACM* (2018). doi:10.1145/3206505.3206527

[71] Cheng-Yao Wang, Wei-Chen Chu, Po-Tsung Chiu, Min-Chieh Hsiu, Yih-Harn Chiang, and Mike Y Chen. 2015. PalmType: Using palms as keyboards for smart glasses. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 153–160.

[72] Cheng-Yao Wang, Min-Chieh Hsiu, Po-Tsung Chiu, Chiao-Hui Chang, Liwei Chan, Bing-Yu Chen, and Mike Y Chen. 2015. Palmgesture: Using palms as gesture interfaces for eyes-free input. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 217–226.

[73] Martin Weigel, Tong Lu, Gilles Bailly, Antti Oulasvirta, Carmel Majidi, and Jürgen Steimle. 2015. ISkin: Flexible, Stretchable and Visually Customizable On-Body Touch Sensors for Mobile Computing. *ACM* (2015). doi:10.1145/2702123.2702391

[74] Frank R Wilson. 1999. *The hand: How its use shapes the brain, language, and human culture*. Vintage.

[75] Wei Wu, Wei Wu, Stephen Elliott, Stephen John Elliott, Sen Lin, Shenshen Sun, and Yandong Tang. 2020. Review of palm vein recognition. *IET Biometrics* (2020). doi:10.1049/iet-bmt.2019.0034

[76] 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*. 85–94.

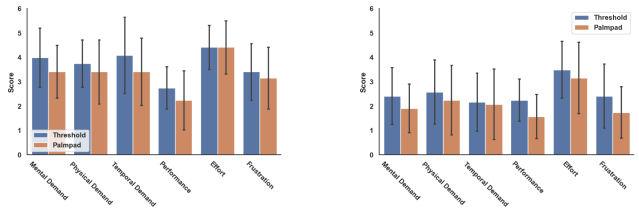
[77] Robert Xiao, Julia Schwarz, Nick Throm, Andrew D Wilson, and Hrvoje Benko. 2018. MRTouch: Adding touch input to head-mounted mixed reality. *IEEE transactions on visualization and computer graphics* 24, 4 (2018), 1653–1660.

[78] Xuekui Yan, Wenxiong Kang, Feiqi Deng, and Qiuxia Wu. 2015. Palm vein recognition based on multi-sampling and feature-level fusion. *Neurocomputing*

(2015). doi:10.1016/j.neucom.2014.10.019
[79] Xing-Dong Yang, Tovi Grossman, Daniel Wigdor, and George Fitzmaurice. 2012. Magic finger: always-available input through finger instrumentation. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. 147–156.
[80] Gu Yizheng, Chun Yu, Zhipeng Li, Zhaoheng Li, Wei Xiaoying, and Yuanchun Shi. 2020. QwertyRing: Text Entry on Physical Surfaces Using a Ring. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* (2020). doi:10.1145/3432204
[81] Sang Ho Yoon, Ke Huo, Vinh P Nguyen, and Karthik Ramani. 2015. TIMMi: Finger-worn textile input device with multimodal sensing in mobile interaction. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*. 269–272.
[82] Xinge Yu, Zhaoqian Xie, Yang Yu, Jungyup Lee, Abraham Vazquez-Guardado, Haiwen Luan, Jasper Ruban, Xin Ning, Aadeel Akhtar, Dengfeng Li, et al. 2019. Skin-integrated wireless haptic interfaces for virtual and augmented reality. *Nature* 575, 7783 (2019), 473–479.
[83] Mubashira Zaman, Soweba Rahman, Tooba Rafique, Filza Ali, and Muhammad Usman Akram. 2017. Hand Gesture Recognition Using Color Markers. In *Proceedings of the 16th International Conference on Hybrid Intelligent Systems (HIS 2016)*, Ajith Abraham, Abdelkrim Haqiq, Adel M. Alimi, Ghita Mezzour, Nizar Rokbani, and Azah Kamilah Muda (Eds.). Springer International Publishing, Cham, 1–10.
[84] Bob Zhang, Wei Li, Pei Qing, and David Zhang. null. Palm-Print Classification by Global Features. *IEEE transactions on systems, man, and cybernetics* (null). doi:10.1109/tsmca.2012.2201465
[85] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. 2020. Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214* (2020).
[86] Yang Zhang, Wolf Kienzle, Yanjun Ma, Shiu S Ng, Hrvoje Benko, and Chris Harrison. 2019. ActiTouch: Robust touch detection for on-skin AR/VR interfaces. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 1151–1159.

A NASA-TLX Scale Results

The results of the NASA-TLX Scale are shown as Figure 12.



(a) Scores for the Fitts's Law task. (b) Scores for the clicking task.

Figure 12: Subjective rating scores for different usability evaluation tasks. 0 - low demand, 6 - high demand.(NASA-TLX)

B Data Samples

Some samples of the dataset are shown as Figure 13.

C Circuit Implementation

We constructed an impedance measurement circuit to generate a label. First, we generated two identical high-frequency signals, denoted as S_{sta} and S_{tes}. The test signal passed through the human body, while the standard signal remained unchanged. Subsequently, we multiplied the two signals, resulting in a signal combined with a high-frequency signal and a low-frequency signal, which we marked as S_{com}.

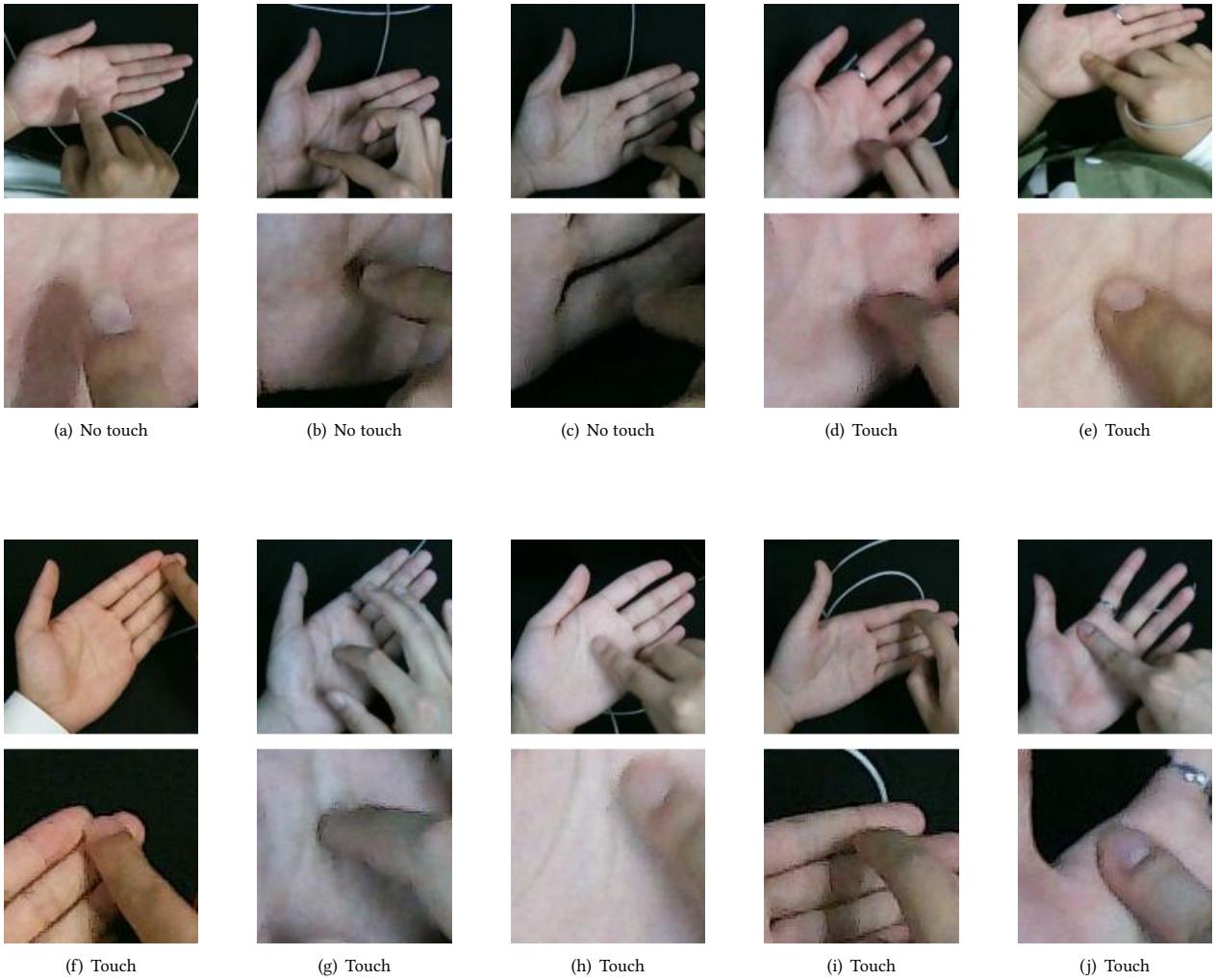
Assuming that the standard signal is $S_{sta} = A_1 cos(\omega t + \phi_1)$ and the test signal is $S_{tes} = A_2 cos(\omega t + \phi_2)$. By multiplying the S_{sta} and the S_{tes}, we get the S_{com}, which is

$$\begin{aligned} S_{com} &= S_{sta} * S_{tes} = A_1 A_2 cos(\omega t + \phi_1) cos(\omega t + \phi_2) \\ &= \frac{A_1 A_2}{2}(cos(2\omega t + \phi_1 + \phi_2) + cos(\phi_1 - \phi_2)) \end{aligned} \tag{1}$$

Afterward, we passed this signal S_{com} through a low-pass filter to extract the low-frequency component. The filtered signal is marked as $S_{lp} = \frac{A_1 A_2}{2} cos(\phi_1 - \phi_2)$

Hence, by filtering, we can obtain the delay between S_{sta} and S_{tes} ($cos(\phi_1 - \phi_2)$), as well as the product of their amplitudes($\frac{A_1 A_2}{2}$). When the right index finger is in different states with the left palm (contact or no contact), there is a noticeable difference in the product of the amplitudes of the two signals. This difference can be used to distinguish the contact state and generate corresponding video labels.

After obtaining the amplitude of the signal S_{com}, we can sample from the signal to get the digital signal. Consequently, we can quantize the amplitude digital signal to obtain a binary representation (0 or 1) of the contact state between the two hands at each moment. This binary signal can then be aligned and downsampled to synchronize with the relatively low-frequency video signal, resulting in labels for each video frame.



(a) No touch (b) No touch (c) No touch (d) Touch (e) Touch

(f) Touch (g) Touch (h) Touch (i) Touch (j) Touch

Figure 13: 10 examples of the first frame from the dataset. The upper image in each example is a hand figure and the lower image is a finger figure. The ground truth of each example is captioned under the figures. All of the prediction labels are same with the corresponding ground truth.