This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3653412

Privacy-Protected Hand Pose Reconstruction and Air Writing via Rolling Spheres

Xiao Zhang, Deniz Acikbas, Soham Naik, Griffin Klevering,
Juexing Wang, Zaynab Mourtada, Li Xiao, Tianxing Li

***Abstract*—Smart homes, medical devices, and education systems, among other emerging cyber-physical systems, hold immense promise for sensing-based user interfaces, especially for using fingers and hand gestures as system input. However, vision approaches compatible with time-consuming image processing adopt low 60 Hz location sampling rate (frame rate) for real-time hand gesture recognition. Furthermore, they are not suitable for low-light environment and long detection range. In this paper, we propose RoFin, which first exploits 6 temporal-spatial 2D rolling fingertips for real-time 3D reconstructing of 20-joint hand pose. RoFin designs active optical labeling for finger identification and enhances inside-frame 3D location tracking via high rolling shutter rate (5-8 kHz). These features enable great potentials for enhanced multi-user HCI, virtual writing for Parkinson suffers, etc. We implement RoFin prototypes with wearable gloves attached with low-power single-colored LED nodes and commercial cameras. The experiment results show that (1) In flexible sensing distances up to 2.5 m, RoFin achieves an average labeling parsing accuracy of 85%, (2) In comparison to vision-based techniques, RoFin improves the tracking grain with 4× more sampled points each frame, (3) RoFin reconstructs a hand pose in real time with 16 mm mean deviation error compared with Leap Motion under flexible distance, and (4) we further investigated real-world applications of RoFin, such as air writing with smoothed trajectories and mobile-based letter/number recognition in our developed *Xamera* app.**

***Index Terms*—Visible light communication, Wearable devices, Gesture recognition, Human computer interaction.**

I. Introduction

Human hands are not just crucial, vital organs for catching and grabbing; they have also long been used for communication, such as in greetings, sign language for the deaf, or hand signs in sports and wars. Hand poses have become direct, and cost-effective Human-Computer Interaction (HCI) across a wide variety of applications due to the fast development of computer technology and artificial intelligence (AI). For example, fingers and hands can be used in smart homes to control IoT devices (e.g., turning devices on/off), interact with video games for a user-friendly and immersive gaming experience (e.g., accelerating race cars), and in XR (AR, VR, and MR) enabled mobile applications to provide interactive operations that are close to reality (e.g., navigation) [1], [2], [3], [4], [5], [6], [7], [8], [9], [10].

Some researchers attach on-body sensors (e.g., accelerators, gyroscopes) to each finger and joint to measure the spatial position variation of fingers. XSens [11] conducts 3D motion capturing enabled by specific and expensive miniature MEMS inertial sensors (e.g., MOVELLA DOT SENSOR price is $132). Other studies utilize wireless signals including radio frequency, sound, and lights (e.g., soli [12], FingerIO [13], and Ali [1]) for hand-free gesture recognition. However, these methods require the expensive or specific devices and have limited sensing distance less than 0.5m.

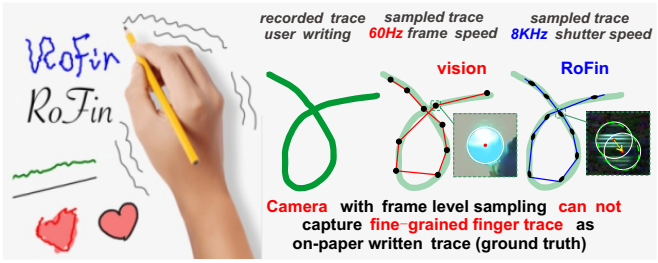


Fig. 1: RoFin with shutter level sampling can better record jitters of hand trace compared with frame level sampling.

The vision-based hand gesture identification approaches are popular, which adopts similar processing as human eyes to detect the morphology of hands with about 60Hz frequency of perception. The accuracy of vision-based hand gesture recognition achieves more than 80% with the help of deep learning [3]. However, vision-based approaches have the following drawbacks: (1) they can not work well in the low light conditions or long detection range due to the limited light amount reflecting from the hand to the camera's image sensor, (2) low sampling rate (e.g., 60 Hz) [14], [15], [16], [17], [18] of cameras when tracking fingers, the same as human eyes with limited perception ability can not see the detailed motion trajectory of Parkinson sufferers' trembling hands clearly, (3) high processing cost and latency because of their recognizing hand morphology with about 20 hand joints, and (4) the captured frame of the scene with hands also poses privacy concerns in sensitive circumstances.

Commercial cameras and LEDs are deployed everywhere, enabling optical camera communication (OCC) a reality in our daily lives [17], [18], [6], [9], [19], [20], [7], [8], [5], [10]. The **rolling shutter** in commercial cameras exposes one row of pixels and generates a whole image row by row. A clear **strip effect** appears when the switching speed of the light wave from the transmitter is equal to or slightly less than the rolling shutter speed. Many researchers have tried to improve data rates by collecting data in rolling strips rather than the entire image frame. However, these systems [21], [16], [22], [17], [8], [23] only exploit rolling shutter for communication instead of sensing such as inside-frame fine-grained location

Xiao Zhang, Deniz Acikbas, Soham Naik, and Zaynab Mourtada are with the department of Computer Information and Science, University of Michigan-Dearborn, USA email: {zhanxiao, dacikbas, sohamn, zaynabmo}@umich.edu. Griffin Klevering, Juexing Wang, Li Xiao and Tianxing Li are with the Department of Computer Science and Engineering, Michigan State University, USA e-mail: {kleveri2, wangjuex,lxiao, litianx2}@msu.edu.

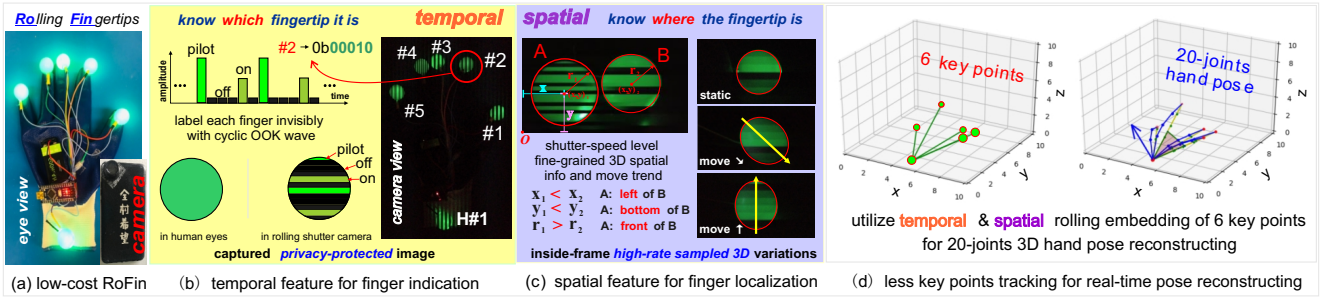


Fig. 2: 3D hand pose reconstructing via 6 temporal-spatial 2D rolling patterns from fingertips and the wrist.

tracking with high sampling rate (rolling shutter speed, e.g, 5 kHz) instead of one sample (1Hz).

In this work, we propose **RoFin**, which is a low-cost and privacy-protected approach for real-time 3D hand pose reconstructing with fine-grained finger tracking ability in flexible distance and varied ambient light conditions. RoFin consists of wearable gloves and a commercial camera, as shown in Figure 2 (a). Each glove finger and the wrist is attached to one low-power LED node controlled by Arduino Nano (<$10). The receiver is the commercial camera (e.g., a smartphone camera). Before employing RoFin, both frame and shutter speeds can be modified manually and quickly using a built-in camera app in professional mode or a third-party app on an Android or iPhone (e.g., Protake). We additionally developed an Android app, *Xamera*, to showcase a use case for RoFin (further details appear in Section VII).

RoFin **first** exploits **2D temporal-spatial rolling** fingertips for ***(1)*** *active optical labeling for fingers/hands*, ***(2)*** *fine-grained inside-frame finger tracking with rolling shutter speed*, and ***(3)*** *real-time 3D hand pose reconstructing*, as shown in Figure 2 (b), (c), and (d) separately. Each LED node covered with same-size sphere emits distinct light waves as optical label invisible to human eyes but perceptive by rolling shutter cameras for robust finger identification. Based on the captured spots (deformed ellipses) via rolling shutter at high sampling rate (e.g., 5 kHz), RoFin can parse fine-grained 3D locations and inside-frame variations of fingertips (left/right, up/down, and front/rear). Finally, RoFin reconstructs 3D hand poses consisting of 20 points by tracking only 6 key points (5 fingertips and 1 wrist point) for less latency and computation overhead. Furthermore, we employed a Kalman filter to continuously smooth the captured trajectories, leveraging fine-grained sampling to compensate for the rolling shutter effect. This enables cross-frame air writing with accurate letter and digit recognition using lightweight CNN models.

There are several commercial approaches closely related with RoFin. Leap Motion [24] utilizes infrared cameras with illumination LEDs to precept hands' morphological variation frame by frame. Luxapose [25] exploits fixed several LED landmarks on the ceiling to interact with rolling camera for indoor localization instead of hand pose reconstruction. In contrast, our RoFin hand gesture recognition has the benefits of: **(1) Less tracking overhead.** Instead of entire hand morphology (i.e., 21 joints), RoFin recognizes a hand with 6 nodes tracking. **(2) Massive optical labeling.** RoFin recognizes multiple hands with identities via high-capacity active optical labeling. **(3) Flexible usage scenarios.** Our RoFin is portable and can work at distance up to 3m in both day and night/dark. **(4) Fine-grained sampling.** RoFin can sample fine-grained 3D fingertips in dynamic at high rolling shutter rate. **(5) Mutual blockage avoidance.** RoFin can avoid most blockage issues because it only uses the fingertips' area rather than the entire hand morphology.

However, we must overcome three **technical challenges**: **C1:** Each finger from multiple hands requires a distinct label identifiable robustly in varied ambient light and long distances. **C2:** It is difficult to decipher the fingertip's fine-grained 3D fluctuation based on the 2D shape (i.e., a distorted ellipse) that was recorded during a frame period. **C3:** We merely track of a hand's 6 key points for reduced overhead. It is a challenge to reconstruct a 20-point 3D hand pose accurately from restricted 6 key points in real-time. **C4:** Fine-grained finger-writing traces tracked from users (including those with Parkinson's disease) tend to be ragged, which hinders the real-time trajectory smoothing needed for digit and letter recognition.

Our **contributions** can be summarized as follows:

(1) RoFin is the first work to exploit rolling shutter effect for 3D hand pose reconstructing via 2D rolling patterns of fingertips. We indicate each fingertip and wrist point with asynchronous cyclic optical labels. Then we adopt a lightweight CNN model with bounding boxes to identify fingertips and wrist points. Our active optical labeling overcomes the limitations of the vision-based technique and is appropriate for the identification of multiple hands in low-light and long-range detection scenarios.

(2) We creatively utilize inside-frame high sampling via rolling shutter to track several fingertips' 3D location variation instead of only one 2D location sample in a frame to enhance tracking granularity further while vision-based approaches only use one 2D location sample during one frame period. The improved finger tracking ability has potentials for the virtual writing for Parkinson's suffers, better user experience for virtual writing/painting in AR/VR/MR, etc. Our RoFin can also be adopted in telesurgery by transferring the fine-grained tracked finger traces of multiple doctors and nurses robots.

(3) Based on the finger identification and parsed 3D location info of 6 key points (5 fingertips and 1 wrist point) from (1) and (2), we design a real-time and lightweight 20-point 3D hand pose reconstructing algorithm HPR from tracked 6 key points. HPR can efficiently reconstruct a 3D hand pose by direct calculation instead of redundancy points' tracking while not sacrificing the reconstructing accuracy. Even a if specific

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3653412

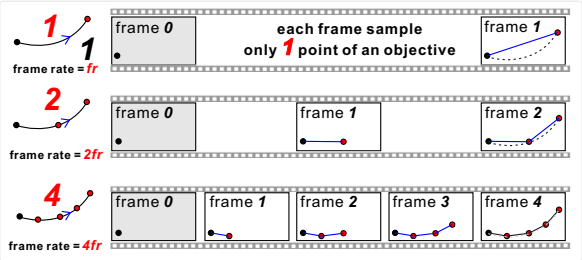


Fig. 3: Frame rate vs. tracking granularity.

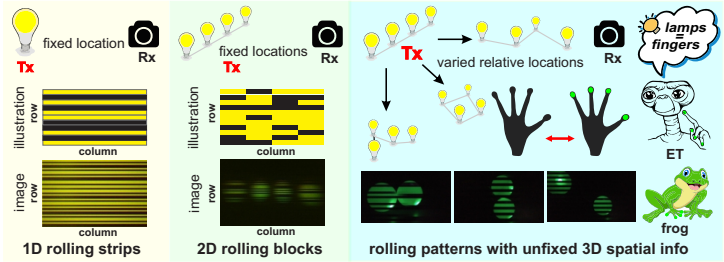


Fig. 4: Strip effect from single to multiple light sources.

fingertip is blocked by others, RoFin can easily infer which finger it is as well as its location due to the fingers relation of a users' hand fixed structure. We also analyze in detail that these six key points can provide sufficient positioning accuracy without incurring additional tracking overhead.

(4) We implemented RoFin on commercial devices and evaluated its performance across three key aspects: (i) finger identification in diverse settings, (ii) in-frame tracking enhancement (compared to vision-based approaches), and (iii) hand pose reconstruction error (with Leap Motion as the benchmark) and reconstruction latency. End-to-end and dynamic-scenario evaluations were also conducted. We further integrated a Kalman filter to smooth finger-writing traces across consecutive frames in our implemented Android app *Xamera*, enabling virtual writing and content recognition to facilitate emergency communications.

II. Background and Related Work

A. Vision-based 3D Hand Pose Recognition

Numerous works adopt cameras to recognize hand poses with computer vision. These approaches can be classified into 2 categories. (1) Hand image searching in pre-computed databases with machine-learning assistance, which capture hand images and then query pre-computed 3D hand models to determine the best-matched hand pose [26], [27], [28]. (2) Calculate 3D coordinates of hand joints directly and then identify the hand pose by optimizing an objective function, which represent the hand with a 3D hand model and adopt an optimization strategy to speed up hand pose prediction [29], [3], [30] However, these existing vision-based hand pose recognition methods are based on complete hand morphology such as hand silhouettes and numerous joints (e.g., 20 joints) with non-trivial tracking and computation overhead.

Furthermore, vision-based approaches sample the location variation at the frame update level shown in Figure 3 while the frame rate is set $\leq$ 60 fps instead of higher to be compatible with time-consuming image processing. In contrast, RoFin enables 3D hand pose reconstruction via incomplete hand morphology, relying on capturing only six 2D rolling spots (5 fingertips and the wrist point) of a hand. With fewer tracking points and a lightweight pose reconstruction algorithm HPR, RoFin achieves the real-time reconstruction of a hand with an average 13.8 ms time cost. Moreover, even with limited 60 fps frame rate, RoFin can sample numerous inside-frame points thanks to the higher rolling shutter rate instead of only 1 in vision approaches, thus enhances the finger tracking granularity which is explained below.

B. Strip Effect in Rolling Shutter Camera

Cameras in our daily-used smart devices adopt low-cost **rolling shutters** to reduce the readout time of pixels from the whole image frame. Rolling shutter camera exposes one row of pixels and generates a whole image row by row. A clear **strip effect** appears when the switching speed of the light wave from the transmitter is equal to or slightly less than the rolling shutter speed, as shown in Figure 4. As a result, optical signals containing transmitted data in a symbol period may be sequentially captured in these rolling strips, which enables OCC (e.g., CASK, ColorBar) [21], [16], [8].

However, the high-rate sampling ability of rolling shutter camera is wasted in current vision-based finger tracking and hand pose recognition. These approaches only sample one location of specific objective (e.g., a fingertip) in a frame despite the rolling shutter camera can capture numerous location samples during a frame period. In contrast, RoFin fully utilizes these numerous location samples to enhance the inside-frame finger tracking granularity via active LED spheres attached on fingertips. The deformed ellipse will be generated when the finger is in high-motion status (e.g., shaking) and records the location variation of centre of LED sphere in 3D space during a frame period, as shown in Figure 2 (c) and 4.

As result, RoFin can improve virtual painting and writing user experience, especially for Parkinson suffers. Parkinson sufferers' tremor frequency (10Hz [31]) indicates how frequently their hands return to the same spot (i.e., the hand shakes from center to left and then to right, then back to the center). Although the frame rate of 60Hz can catch one cycle of tremor with 60Hz/10Hz=6 sampled points, RoFin samples more points at higher shutter speed (e.g., 10 kHz). Considering these tremor cycles with random 3D motion routes, more sampling points for each cycle are indeed needed for fine-grained random tremor trajectories tracking and further trace optimization rather than vision-based coarse-sampled traces.

III. RoFin System Overview

In this section, we briefly introduce the composition, workflow and 4 main tasks of RoFin, as illustrated in Figure 5.

Composition. RoFin system consists of two parts. **(1) RoFin gloves** are commercial insulating gloves where each fingertip and the wrist are attached with a low-power LED component covered with a plastic ball. These LED components are controlled by an Arduino Nano to generate identical LED waves to indicate different fingertips. **(2) RoFin reader** is based on commercial cameras (e.g., smartphones, web cameras, etc). These cameras use adjustable focal length lenses and rolling shutters with configurable shutter rates.

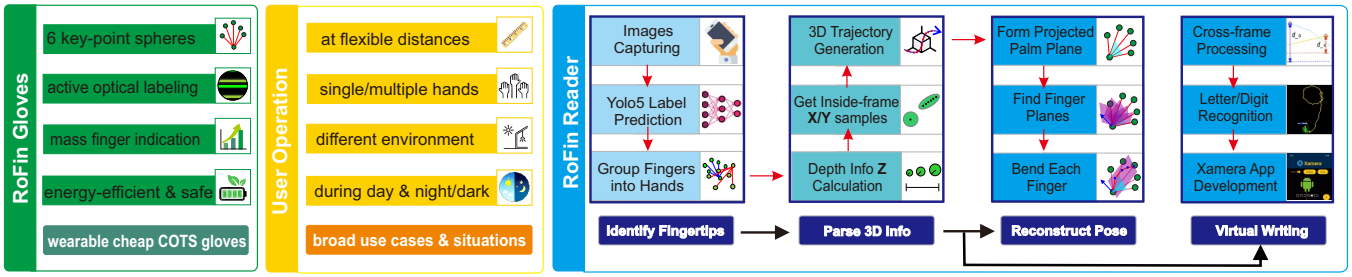


Fig. 5: RoFin system overview: composition, workflow and four main tasks.

Workflow. **(i)** The user puts on RoFin gloves and makes some hand poses. **(ii)** After setting the rolling shutter rate and focal length, RoFin reader captures the continuous 2D rolling spots of six key points (5 fingertips and 1 wrist point) frame by frame. **(iii)** RoFin reader identifies each fingertip/wrist point via lightweight CNN model with bounding boxes. **(iv)** RoFin parses the 3D location variations of each key point based on captured deformed ellipses in each frame with the granularity of strip width. **(v)** Then, RoFin reconstruct 3D hand pose via lightweight HPR algorithm based on the parsed label and its fine-grained 3D location. **(vi)** Meanwhile, for the virtual writing application, RoFin employs a Kalman filter to further smooth cross-frame fine-grained writing trajectories sampled under rolling shutter effects.

Four Main Tasks. At a high level, RoFin addresses two core questions: **(1) identifying which fingertip** (*Section IV*) it is, and **(2) locating the position and in-frame variation** (*Section V*) of this fingertip at a sampling rate matching the rolling shutter speed. RoFin further **(3) reconstructs the 3D hand pose** (*Section VI*) via the HPR algorithm, using outputs from tasks (1) and (2), and **(4) generates smooth trajectories** that are then recognized as letters/digits (*Section VII*) through our developed Android app *Xamera*.

IV. Active Optical Labeling

A. Temporal Rolling Patterns

The light source emits optical signals varied with time sequences at rolling shutter speed level during one frame period can be recorded row by row in the captured image frame by the rolling shutter camera, as illustrated in Section II-B. In our prototype, the Arduino Nano allows us to control and configure the ON/OFF switching rate of the LEDs by running executable codes onboard. The shutter rate is configured manually on smartphone. We set the rolling shutter rate (i.e., the parameter `shutter speed`) via the professional mode of the system camera for Android smartphones, the Protake app for iPhones, and the Camera 2 API which we called to configure it in our developed *Xamera* App.

Only when the rolling shutter rate is similar to the transmission frequency, however, can we clearly see the distinct rolling strips, as illustrated in Figure 6 (a). We can utilize captured rolling spots with distinct strip textures as active optical labels to **indicate fingertips**. However, optical signals have multiple light features varied with temporal sequences such as different amplitudes, transmission frequencies or colors can be used for indication of different fingertips. Which ought to be used in rolling patterns for RoFin?

Our design exploration is shown in Figure 6 (b). **(1) Amplitude.** We can adjust brightness of the light source with time sequences[32], [33], [34]. The light amplitude fluctuation is vividly captured sequentially. **(2) Transmission Frequency.** We may also alter the ON/OFF switching speed of the light wave. **(3) Color Spectrum.** We could transmit the light with different wavelengths. The captured rolling strips are colorful and vary in the same way of color fluctuation with time sequences as the light source does.

Our choice: Amplitude. It requires RGB LED and complicated modulation to achieve **color spectrum** diversity. Complex modulation and a longer time period to present complete frequency variation (i.e., only partial of the complete pattern could be presented on the captured spot of sphere with limited width) are both necessary for **transmission frequency** diversity. To indicate multiple fingertips, **amplitude variation** is more suitable compared with different colors or transmission frequency which require complex devices (i.e., multi-color LED, high-clock-rate MCU) and control overhead. To create a low-cost design while remaining robust, RoFin uses single-color (i.e., green, or other color options) LEDs and Pulse Width Modulation (PWM) based amplitude configuration.

B. Fingertip and Hand Indication

Each attached LED element can emit the different amplitude waves as the active optical labels. However, we can not syn-

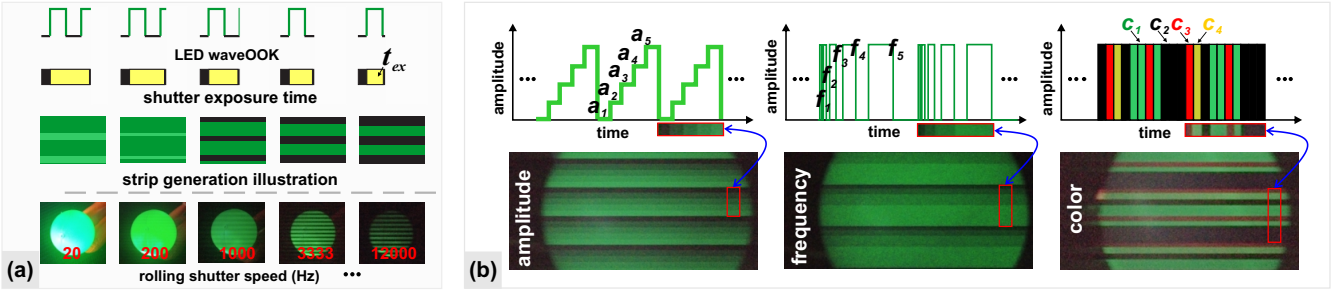


Fig. 6: (a) Captured strips impacted by shutter speed. (b) Light feature selection for temporal rolling patterns.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3653412

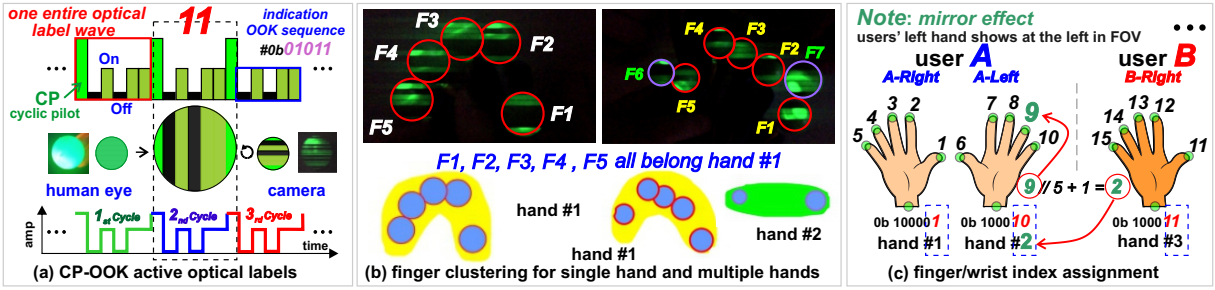


Fig. 7: The scheme design of CP-OOK fingertip indication and wrist point labeling. Based on parsed indication number of fingertips and wrist points, the different hands are clustered for further finger tracking and hand pose reconstruction. The indication capacity of RoFin is flexible and massive by easily adjusting the number of the indication sequence.

chronously control each light source to let them start temporal rolling pattern at the same time. Additionally, because of their various positions inside the field of view (FOV) of the camera, the recorded rolling strip may begin at a different time. These asynchronous problems make it difficult for the RoFin reader (i.e., camera) to recognize the embedded identification information from different light sources (i.e., LEDs). Thus, we design asynchronous Cyclic-Pilot On-Off-Keying (CP-OOK) labeling scheme for different fingertips from multiple hands and wrist-assisted hand indication.

1) CP-OOK based Fingertip Indication:

The optical label consists of two parts: **(1) CP (cyclic pilot)**, takes one symbol period at the beginning, and **(2) indication sequences**, formed via 5 (can be extended) OOK (On-Off Keying) symbols, as shown in Figure 7 (a). Aside from the Off symbol (dark), the optical label design has two amplitude levels: the CP symbol has the highest brightness, whereas the On symbol has the lower brightness of CP.

Instead of the normal very long preamble [35], we designed a short pilot (i.e., CP). Because the number of rolling strips revealed in the finger pattern (the circle or ellipse) is restricted, we must ensure that at least one complete optical label is shown in each rolling pattern for further decoding. Furthermore, to improve the robustness of these optical labels in variable environment, we set a total of 2 non-dark amplitudes ($\mathtt{Am}_{CP}$, and $\mathtt{Am}_{On}$) instead of additional amplitude levels (e.g., 5 amplitude levels in amplitude shift keying).

We encode the index of each finger with its binary number into OOK symbols, as shown in Figure 7 (a). When the finger index is 11, for example, the binary number is *01011* and the indication sequence is [*Off, On, Off, On, On*]. The length of the indication sequence is determined by the number of fingers being tracked. 3 OOK symbols can represent up to 8 fingers, enough for 1 hand. 4 OOK symbols can represent 16 fingers, enough for 3 hands. In general, N OOK symbols can represent 2^N fingers that are appropriate for $2^N/5$ hands. The transmission frequency of light waves have the same or slightly slower frequency than the rolling shutter, and thus these optical labels are clearly recorded for further finger identification, as shown in Figure 7 (b). RoFin has great potential of massive indication capacity. The number of indication sequences can easily be adjusted by setting proper rolling strip width (i.e., configure transmission frequency and shutter speed). For example, N can be set to 7 to indicate 2^7 = 128 fingers of 12 users. In contrast, Leap Motion is not extendable and is unable to distinguish between the hands of various players in multi-user games and HCI. For instance, 4 Leap-Motion users pay 4 × 250 = 1000, while 4 RoFin users cost \$150 (\$50 of a commercial camera plus 4 × \$25 of gloves).

2) Wrist-assisted Hand Indication:

We assign each finger from multiple hands of multiple users with a finger index as illustrated in Figure 7 (c). Given users A, B, and so on. We assign the A's right hand as the hand #1, A's left hand as hand #2, we assign the B's right hand as hand #3, and the rest can be done in the same manner. In this paper, we evaluate three hands (A's right hand and left hand, B's right hand). We assign these fingers with indication index from 1 to 15 finger by finger as shown in Figure 7 (c).

However, only 5 fingertips are not enough to determine a hand in a 3D space. Besides five fingertips of a hand, we also attach an LED node covered with same-size sphere at the end of the wrist. This additional wrist point has more vital meaning for hand pose reconstructing in comparison to other five fingertips (discussed in Section V). Furthermore, different hands should have distinct indications for these 6 key points of each hand (5 fingertips and 1 wrist point) for correct reconstructing each hand pose when they are shown in the camera view at the same time.

Based on the analysis above, the indication of the wrist should have more significant indication than the fingertips but not introduce additional non-trivial overhead (e.g, use different light features: colored-LED, different modulation schemes: FSK, etc). To achieve this design goal, we use the same CP-OOK modulation in fingertip indication, but set the **leftmost** indication bit as **1** while the remaining bit sequence as the wrist indication for differentiation.

Given three hands #1, #2, and #3, it requires 4-bit indication sequence to denote 15 fingers. Thus, originally, the binary number for finger #11 is `1011`. But we set its indication sequence as **`0`**`1011`, which is [***Off***, *On, Off, On, On*], to make it compatible for wrist point indication. For the wrist point from #2, its binary number is 10. Following the rule above, the indication sequence of this wrist point is set as `100`**`10`**, which is [***On***, *Off, Off,* ***On***, ***Off***].

3) Impact of OOK Length on Latency:

Both finger indication and wrist indication rely on LED wave-generated rolling strips displayed on plastic spheres. Consequently, two key questions arise: (1) How many fingers and wrists can a RoFin system support? (2) How many strips can a single sphere present?

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3653412

Three main factors influence these outcomes: (1) Transmission frequency and shutter rate: Faster LED wave transmission frequencies and corresponding shutter rates generate shorter rolling strips, enabling more strips to be displayed. (2) Sphere size: Larger spheres offer greater vertical width, allowing for more rolling strips. (3) Distance and lens settings: Increased distance between a sphere and the camera reduces the sphere's apparent size in the image, resulting in fewer displayed strips. However, this distance limitation can be mitigated by adjusting the smartphone's focal length. To determine the maximum capacity of our current RoFin technology, we conducted strip count tests at a typical usage distance of 1.5 m. We used spheres with a 2 cm radius and adjusted the focal length to ensure the hand was properly imaged. As shown in Figure 8, a single sphere can display 21 strips in total. Given the system's ability to accommodate two complete CP-OOK indication sequences, the maximum indication length supported is 21 // 2 = 10. Excluding the CP and wrist/finger indication bits, 8 bits are available for finger identification, resulting in a total of 2^8 = 256 unique finger identifiers.

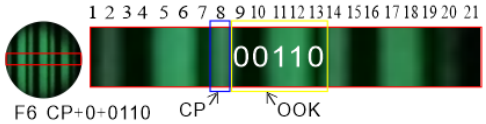


Fig. 8: The number of presented strips at setting of 1.5m with proper captured hand size when set proper focal lens.

Another concern is about the latency or overhead difference of the various OOK lengths. We conduct experiments with a transmission frequency of 8 kHz and the corresponding shutter rate. We measure the time cost of generating LED waves with different indication lengths for fingers from the current setting 4 to the maximum 8. As shown in Figure 9, there is no significant time cost difference among different OOK length settings for finger indication with the average time cost of 1 ms per cycle of a complete CP-OOK indication sequence.

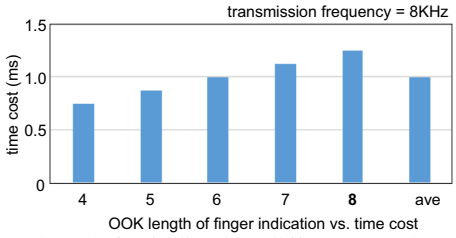


Fig. 9: OOK length for indicating different numbers of fingers and wrists vs. time cost of LED wave generation.

V. 3D Spatial Parsing

Although vision based approaches can use higher frame rate (e.g., 120 fps or 240 fps) for sampling, the image processing is still time-consuming. Thus, vision based approaches can not achieve fast hand pose reconstructing with the same speed of faster frame rate. Thus, vision based approaches normally set the frame rate at about 60 fps for real-time user experience. Different from vision based approaches which only use one 2D location sample (x, y) in each image frame, RoFin has two specific abilities (Note: cameras in RoFin can be set with fixed location or in mobile):

(1) Ability of tracking more 2D location samples (*multiple X and Y values of the center point of the sphere in motion, i.e., inside-frame X/Y sampling*) in each image frame thanks to its sampling at shutter rate instead of frame rate. This ability can enable RoFin's **first** core function: **finer-grained finger's tracking (Section V-A)**. We introduce finer-grained X/Y tracking inside one frame and among continuous frames in Section V-A1 and V-A2 separately.

(2) Ability to reflect depth info Z besides X and Y (Z value: from the sphere to the camera) via perspective principle based on the captured sphere's size variation. Combined with a YOLO's bonding box parsed one X value and Y value (*the center point of the sphere, i.e., inter-frame X/Y sampling*) in each image frame at 60 fps frame rate, the pair of X,Y, and Z values can be used as the **3D location input** for RoFin's **second** core function: **real-time 3D hand pose reconstruction (Section VI)**. We present YOLO enabled rolling label identification and inter-frame X/Y position in Section V-B1 and Z calculation in Section V-B2.

A. Finer-grained X/Y Tracking

1) Inside One Frame:

Why high-rate inside-frame sampling? The objectives are mostly in mobile with random trajectory in real-life situations (e.g., vehicles, drones, or fingers). For example, it is required for numerous location samples in unit time to recover the real trajectory of fingertip as brush in virtual writing/painting. Either a long, random curve that is drawn quickly or a small curve requires more samples to capture more details. However, existing vision-based approaches sample the location variation at the level of frame update. Besides, the frame rate is set to about 60 fps instead of higher frame rate considering the time-consuming image processing. To break this gap, we creatively propose to utilize the rolling shutter effect for numerous inside-frame location samples.

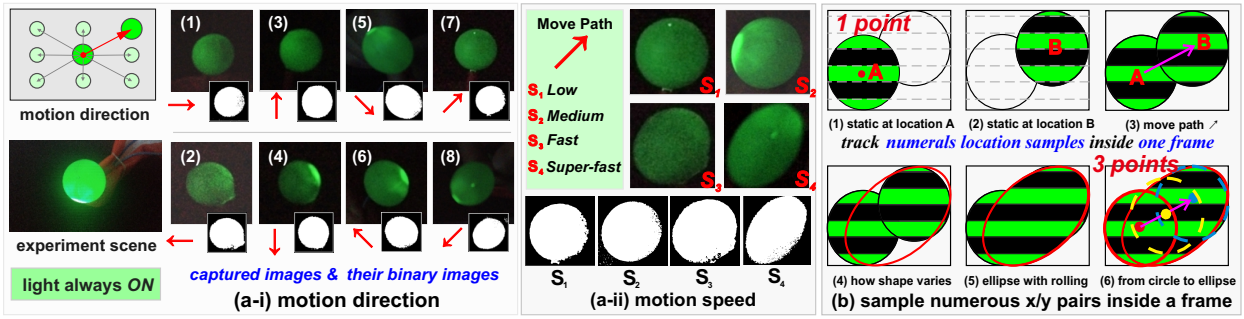


Fig. 10: (a) Impacts of the shape variation of deformed ellipse: (i) motion direction and (ii) motion speed. (b) The sphere center's location variation is recorded in the deformed ellipse with the granularity of strip width.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2026.3653412

Impact of Motion Direction. We move the light source with different directions with constant distance from the light source to the camera plane and the motion speed. For example: (1) and (2) from left to right (→) and reversed (←); (3) and (4) from bottom to top (↑) and reversed (↓); (5) and (6) from upleft to bottomright (↘) and reversed (↖); and (7) and (8) from bottomleft to upright (↗) and reversed (↙). As shown in Figure 10 (a-i), the captured spot shape changes to an ellipse rather than the previous circle and its long axis can reflect the moving direction of the light source.

Impact of Motion Speed. We set 4 levels of motion speed of the light source (i.e, low, medium, fast, and super-fast) with the same motion direction (↗) and fixed distance to the camera plane. As the motion speed increases, so does the length of the ellipse's long axis, as shown in Figure 10 (a-ii).

Numerous Inside-frame X/Y Location Samples. The captured circle or ellipse's pixel index range in columns and rows reflects the horizontal and vertical location information independently. The circle shape means the fingertip/wrist point is not moving or moving slow in the image plane during the entire frame period, and its center location (x, y) can be treated as its location in horizontal and vertical directions. The deformed ellipse records the detailed inside-frame motion with the sample rate at rolling shutter speed, as illustrated in Figure 2 and Figure 11 (b).

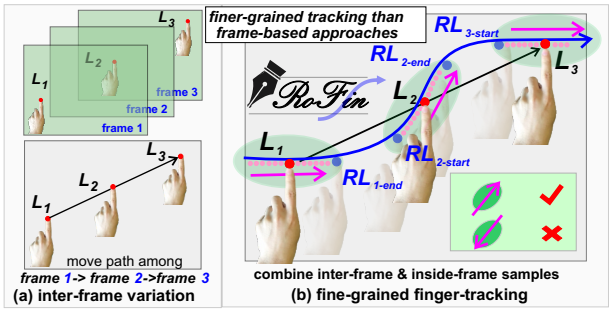


Fig. 11: RoFin's finger tracking among frames combined with numerous inside-frame samples.

2) Among Continuous Frames:

As shown in Figure 10 (a-i), the opposite moving direction of the light source has the same pattern shape (i.e, the ellipse with similar long axis direction). There are 3 frames in Figure 11 (a): `frame1`, `frame2`, and `frame3`. In `frame2`, the light source may move with possible trends as (↗) or (↙) and thus we can not determine fingertip's motion with separate frame.

Moving Trend Determination. However, if we combine the inside-frame moving direction candidates with two continuous frames, we can know the finger's moving trend. Because these frames are continuously generated with time sequences, the end position of finger pattern in previous frame will be close to the start position of finger pattern as shown in Figure 11 (b). Thus, we can determine the finger's moving trend by finding the closest positions of finger pattern in two continuous frames. In this example, the position point RL_{1-end} and $\text{RL}_{2-start}$ are the closest position points between two continuous frames `frame1` and `frame2`.

Moving Trajectory Generation. More importantly, the moving trend determination method is a one-time initialization phase that only requires one frame duration to determine the end positions of each finger pattern and record them as the start positions for the next frame. In this example, using the finger pattern position in `frame3`, we can know the point $\text{RL}_{3-start}$ is the start point. Then we can track finger locations by combining these numerous inside-frame samples and updating them frame by frame, as illustrated in Figure 11 (b). Finally, we can generate a finer-grained moving trajectory in RoFin than the vision-based approach.

B. 3D Location Pair as HPR Input

1) Rolling Label Identification and X/Y Parsing via YOLO: Traditionally, we could decode these optical labels based on the amplitude thresholds via computer vision tools. However, due to the variable optical environment, it is difficult to configure the thresholds dynamically. Even in the same ambient light settings, the captured rolling pattern for each finger requires different thresholds for decoding. Convolutional Neural Networks (CNN) are widely applied in computer vision object classification due to their great robustness and accuracy. The benefits include: (1) Offline training and online identification can reduce latency for real-time finger label parsing; (2) even in high ambient light and difficult to distinguish CP and On, the CNN model can learn the features in the repeating dark and bright rolling strips.

We adopt YOLOv5 for optical label identification with related bounding boxes. YOLO (You Only Look Once) models are commonly used for objects detection since their fast inference with high accuracy. The network structure of YOLOv5 consists of EfficientNet backbone structures, BiFPN

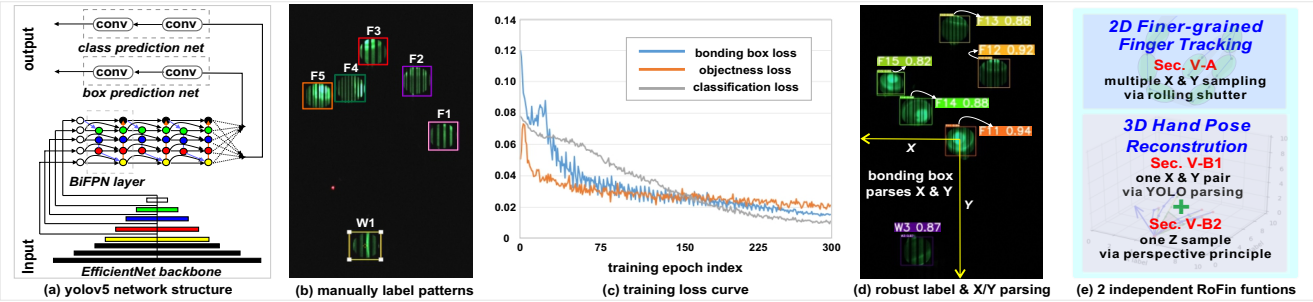
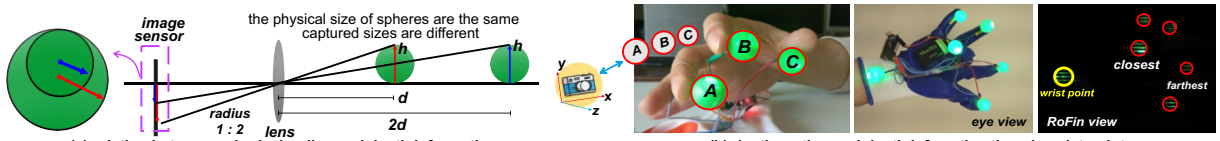


Fig. 12: (a) The adopted CNN network architecture. (b) Example of manually label key points. (c) The training loss curves of bonding box loss, objectness loss (confidence of object presence), and classification loss with 300 epochs. (d) Example of YOLO parsed key point label and X/Y pair. (e) Two independent functions in RoFin and related section content.



(a) relation between spherical radius and depth information (b) depth sorting and depth info estimation via wrist point

Fig. 13: Absolute depth calculation via perspective principle by using wrist point as the reference.

(Bi-directional Feature Pyramid Network) layers to extract object's features effectively, as shown in Figure 12 (a). Then these features are fed through the prediction nets for both objective's class and location of boxes as output.

We capture 90 images of 3 RoFin gloves in 3 different ambient light strengths with 10 images for each setting. Then, we manually label each rolling pattern with 18 class labels (i.e., F1-F15, W1-W3). One example is as shown in Figure 12 (b). In the training, we assign 62 images in the Train dataset, 18 images in the Valid dataset, and 10 images in the Test dataset. We adopt data augmentation via the gray-scale modification to 60% of images and output 3 images per training example to increase the size of training dataset.

As shown in Figure 12 (c), the training losses decrease significantly after 300 epochs of training. We present 3 different training loss results: (1) bonding box loss, which demonstrates that generated bonding boxes are accurate, (2) objectness loss, which indicates the confidence of prediction for object presence in the image, and (3) classification loss, which can reflect the accuracy of classification of different fingertips/wrist points. Finally, we use the trained model to infer the rolling pattern's label with bounding boxes. As an example shown in Figure 12 (d), RoFin outputs labels accurately with high confidence. Besides, these outputted bounding boxes include each sphere's x,y, and radius, which are used for further hand pose reconstructing in Section VI.

2) Depth Info Estimation: Z:

Perspective Principle. We keep light source fixed in the FOV while moving the light source closer or farther to the camera. The captured spot grows and shrinks separately due to perspective principle. Then, we could calculate the depth info Z (i.e., the front and back), as shown in Figure 13 (a).

Absolute Depth Calculation. The wrist point is designed not only for assistance for hand indication illustrated in Section IV-B2, its captured diameter ϕ_w (unit: pixel) can also be used to calculate the absolute distance of key points d to the camera. As shown in Figure 13 (b), the distances to the camera has the relations: $\frac{1m}{d} = \frac{\phi_w}{\phi_{1m}}$. Thus, the absolute distance d from the wrist point to the camera can be formulated as $d = \frac{\phi_{1m}}{\phi_w}$. To do so, we measure and store the captured spot diameter of wrist point at 1m as reference for depth info estimation of all six key points using the same manner.

Coordination Transformation. We set the center of wrist point is the origin of 3D coordinate system. As shown in the right of Figure 2, the z value of the five fingertips is set as their physically relative depth distance value to the wrist point. The center (x, y) of each fingertip's spot shown in the image plane is the pixel value which we need to convert to the physical distance as well. We also use the pixel value range of the wrist point's diameter which maps to the 19 mm of the plastic sphere as the reference to convert the relative X/Y value of each fingertip's center into their relative physical distance to the wrist point separately.

VI. Hand Pose Reconstructing

A. Cluster Fingers and Wrists into Hands

Grouped 6 Key Points of a Hand. Based on the identified fingertips and wrists from multiple hands above, we can calculate their hand belonging separately. And then we can easily cluster fingertips and wrist points from one hand together. For example, the fingers which have indication numbers in [1, 2, 3, 4, 5] and the wrist point with an indication number of 1 should be grouped in hand #1 due to their calculated hand index being the same, which is 1. As shown in Figure 14 (a), the wrist labeling avoids the wrong finger clustering with the wrist point from another hand and thus guarantees further accurate hand pose reconstruction.

3D Coordinates of 6 Key Points. The 6 key points with 3D coordinates clustered into one hand will be input into the HPR model and then the HPR model outputs the reconstructed 3D hand pose in real time. Different from fine-grained finger tracking with numerous inside-frame sampled points in an image frame (Section V), the real-time hand pose reconstructing requires only one 3D location sample for each of six key points per frame for processing.

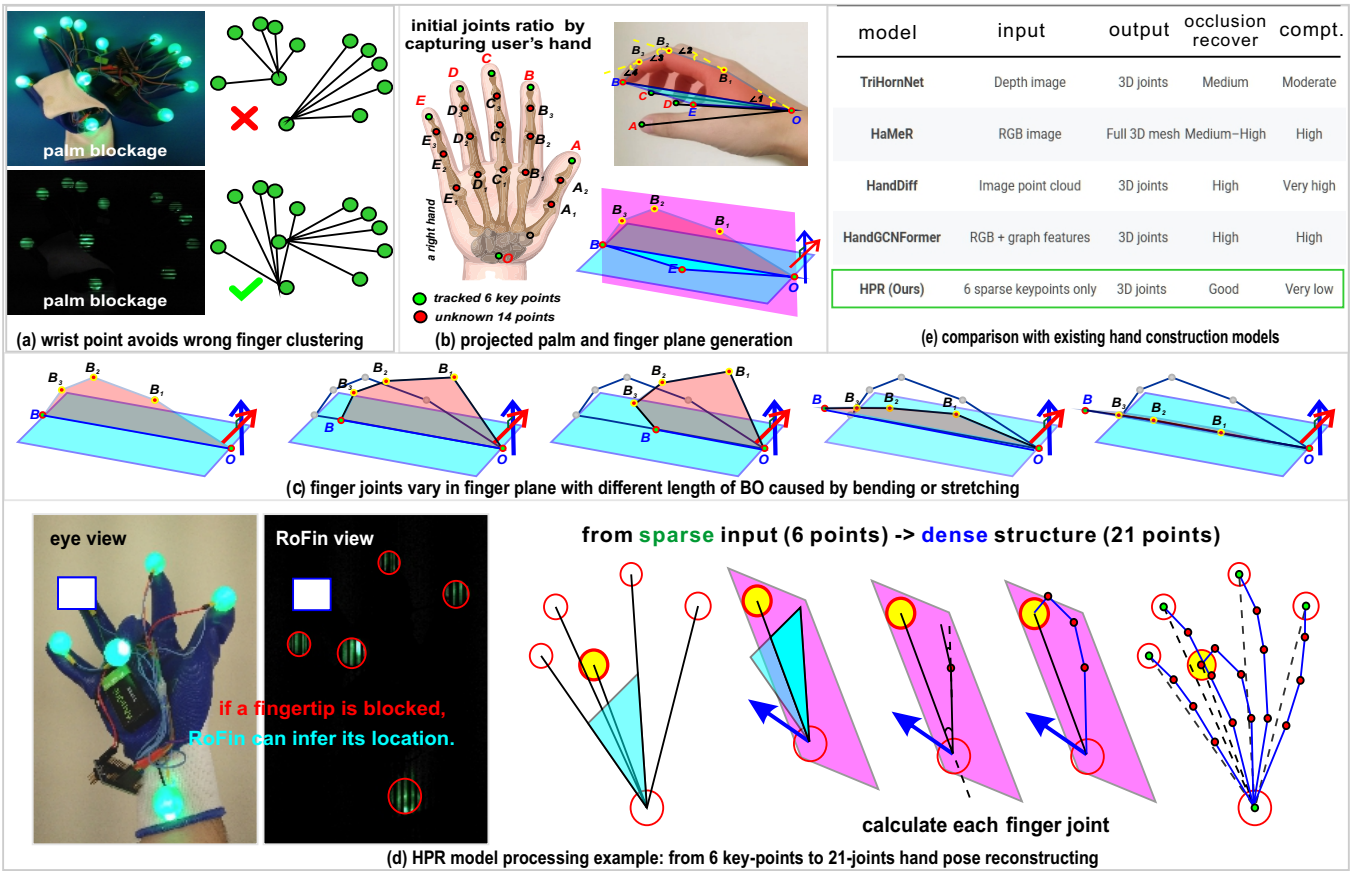
B. Lightweight HPR Model

RoFin is initialized by taking one/several photos of a specific user's open and unobstructed hand to obtain joint ratios before the usage, as shown in the left of Figure 14 (b). Our goal is to calculate the unknown joints according to 6 key points with 3D coordinates (the wrist point p_O, the tip of thumb p_A, the tip of index finger p_B, the tip of middle finger p_C, the tip of ring finger p_D, and the tip of little finger p_E), as shown in Figure 14 (c). However, how can we reconstruct a 20-joints hand pose? The intuitive answer is to calculate 3D locations of the other 14 points: p_{A_1}, p_{A_2} (i.e., we simplify the thumb finger with 2 joints), p_{B_1}, p_{B_2}, p_{B_3}, p_{C_1}, p_{C_2}, p_{C_3}, p_{D_1}, p_{D_2}, p_{D_3}, p_{E_1}, p_{E_2}, and p_{E_3}.

1) Projected Palm and Finger Planes:

The Plane of Projected Palm. As shown in Figure 14 (b)-(d), fingers and palm are projected on the plane defined as projected palm P_{palm}. Actually, the index finger tip, the little finger tip and the wrist point consist of the P_{palm} (i.e., P_{BOE}).

The Plane Formed by Finger Joints. The joints of a finger form a finger plane. These finger planes (except the thumb finger plane) are perpendicular to the plane P_{palm}. For example, joints of the index finger: p_{B_1}, p_{B_2}, p_{B_3}, p_B, and the wrist point p_O generates the finger plane $P_{OB_1B_2B_3B}$. And $P_{OB_1B_2B_3B} \perp P_{palm}$ (i.e., $P_{OB_1B_2B_3B} \perp P_{BOE}$). In contrast to finger planes above, the thumb finger plane is almost parallel to the plane P_{palm} (i.e., $P_{OA_1A_2A} \perp P_{BOE}$). Thus we can



model	input	output	occlusion recover	compt.
TriHornNet	Depth image	3D joints	Medium	Moderate
HaMeR	RGB image	Full 3D mesh	Medium–High	High
HandDiff	Image point cloud	3D joints	High	Very high
HandGCNFormer	RGB + graph features	3D joints	High	High
HPR (Ours)	6 sparse keypoints only	3D joints	Good	Very low

(e) comparison with existing hand construction models

Fig. 14: HPR hand pose reconstruction: (a) wrist point avoids wrong finger clustering, (b) projected palm and finger plane generation, (c) finger joints vary in finger plane with different length of BO caused by bending or stretching, (d) HPR model processing example, and (e) comparison with existing hand construction models [36], [37], [38], [39].

find these 5 finger planes based on the known plane P_{BOE}, as shown in Figure 14 (b)-(c).

2) Bending Fingers in Finger Planes:

Given the 5 connection lines between each fingertip to the wrist point (i.e., l_{OA}, l_{OB}, l_{OC}, l_{OD} and l_{OE}) and the calculated finger planes $P_{OA_1A_2A}$, $P_{OB_1B_2B_3B}$, $P_{OC_1C_2C_3C}$, $P_{OD_1D_2D_3D}$, and $P_{OE_1E_2E_3E}$, we can determine the unknown 14 joints (underlined, the red joints in Figure 14 (b)) on the finger planes via two rules. **R1:** We simplify finger bending because each finger section from one finger bends with a same or proportional angle. **R2:** The length from fingertips to the wrist l_{con} equals to the sum of each finger section's projection to the line l_{con}. We can calculate the bending angle and further find each unknown joint location.

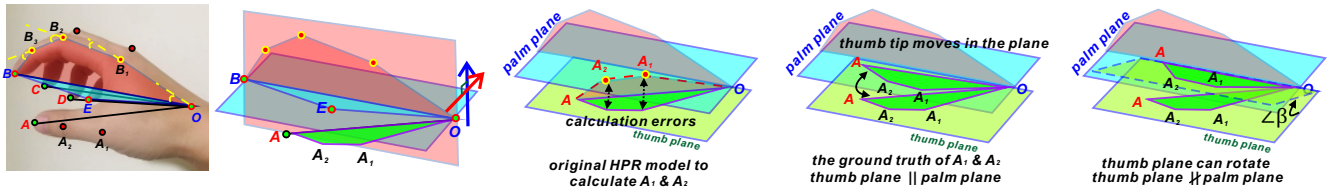
As shown in Figure 14 (c), the finger joints of the index finger vary in its finger plane with different lengths of l_{con} (i.e., l_{OB}). Thus, given a value of variable l_{con}, the 3D locations of other joints from this finger are fixed and can be calculated. For example, we know the length of each finger section of the index finger (i.e., l_{OB_1}, $l_{B_1B_2}$, $l_{B_2B_3}$, and l_{B_3B}) by the initial measurement step. Given the calculated l_{OB} (i.e.,l_{con}) from Section V, the unknown bending angle for index finger $\angle\alpha$ can be calculated by the equation below:

$$l_{OB_1}\times\cos 2\underline{\alpha} + l_{B_1B_2}\times\cos\underline{\alpha} + l_{B_2B_3}\times\cos\underline{\alpha} + l_{B_3B}\times\cos\underline{\alpha} = l_{OB}.$$

3) Calculation Error for Thumb's Joints:

As illustrated in Figure 15 (a) and (b), the thumb plane (AA_2A_1O) is mostly not perpendicular to the palm plane unlike other four fingers and thus resulting calculation errors for joints A_1 and A_2 when using the aforementioned HPR model. The thumb exhibits two types of motion: (1) the thumb plane is parallel with the palm plane and the thumb tip moves in the thumb plane (as shown in the left of Figure 15 (c)); (2) the thumb plane rotates by a small angle $\angle\beta$ with the palm plane (as shown in the right of Figure 15 (c)).

To further improve the localization of joints A_1 and A_2, we can set the proper value of $\angle\beta$ (e.g., 30°) and calculate A_1 and A_2 based



(a) the thumb plane illustration (b) the calculation error (c) two moving types of thumb joints

Fig. 15: The calculation errors with the HPR model for the joints A_1 and A_2. Two types of thumb plane's motion: (1) thumb tip moves inside of the thumb plane, (2) the thumb plane rotates with $\angle\beta$.

on space geometry relation. Besides this rough optimization, we can also add A_1 or A_2 or both A_1 and A_2 as the additional key points to directly determine the thumb plane and then know the locations of thumb joints A_1 and A_2.

4) Tradeoff between Key Points and Accuracy:

There are multiple options for the number of tracked key points to reconstruct a hand, which represents a tradeoff between hand pose reconstruction accuracy and tracking cost/overhead. For example, if we track A,B,C,D,E,O and additional joints such as joints A_2 of thumb, E_2 of the little finger in the top left of Figure 14 (b), hand pose reconstruction accuracy will be significantly improved by avoiding the calculation errors associated with the simplified R1 and R2 rules, as illustrated in Section VI-B3. However, more accurate reconstruction with more tracked points means more tracking overhead. The optimal number of tracked joints depends on the specific application objectives and precision requirements.

We analyze this tradeoff between the overhead caused by the number of key points and the achieved accuracy for our HPR model in detail. There are three types of additional tracked joints: `set1` (the joints of the thumb A_1, A_2), `set2` (the joints of B_1, C_1, D_1, and E_1), and `set3` (the joints between joints in `set2` and the corresponding finger tips, i.e., B_2, B_3, C_2, C_3, D_2, D_3, E_2 and E_3). Regarding the points in `set1`, direct tracking of these joints significantly improves localization accuracy. For `set2`, the additional tracked joints not only enhance the localization accuracy of `set3` joints but also achieve higher precision through their own direct tracking. However, for `set3`, the improvement in localizing `set1` and `set2` joints is limited, as the HPR model already calculates their locations accurately.

5) Hand Model Comparison, Blockage and Limitation:

Similar to the standard IK (inverse kinematic) model [40], which determines parameters to control each joint of the end effector (e.g., robot hand or arm) to present a specific gesture in a physical or virtual 3D space, RoFin significantly simplifies the calculation using two geometric rules instead of the Jacobian matrix, which considers joint speed and motion power. In contrast to IK, which computes joints sequentially from the root to the fingertip, RoFin calculates middle joints directly based on the wrist and fingertip positions. Recently, 3D hand pose and mesh estimation has been advanced by methods using dense inputs such as RGB images, depth maps, or image point clouds with deep models to predict full joint sets or meshes. For example, TriHorn Net estimates 3D hand pose from depth images by predicting 2D joint locations and depths, achieving high accuracy and fast inference [36]. HaMeR reconstructs full 3D hand meshes from a single RGB image using a transformer-based architecture and is robust under varied appearances and occlusions [37]. HandDiff formulates pose estimation as a generative diffusion process conditioned on image point cloud input, improving localization under challenging conditions [38]. HandGCNFormer combines topology-aware graph decoding with transformer features to handle self-occlusion and ambiguous poses [39]. While these methods achieve high fidelity, they require dense high-dimensional inputs and substantial computation, and continuous visual capture may raise privacy concerns. In contrast, RoFin HPR uses only six 3D keypoints (i.e., the wrist and five fingertips) plus a one-time per-user calibration of joint length ratios. By applying simple geometric rules, palm plane projection, finger bending planes, and segment projection constraints, RoFin can reconstruct a full 3D hand pose (up to 21 joints) with minimal overhead. This enables real-time operation on lightweight hardware and allows intuitive inference of partially occluded fingers, as spatial relationships among visible keypoints constrain hidden joint locations. In the future, this occlusion recovery capability could be further enhanced by integrating ideas from dense-input methods such as HandDiff, HaMeR, and HandGCNFormer, achieving more robust estimation under severe occlusion while maintaining the privacy-preserving advantage of sparse, non-visual input. RoFin thus offers a practical, efficient, interpretable, and privacy-conscious solution for resource-constrained or contactless sensing applications, including wearable interfaces, AR/VR, and monitoring systems.

VII. Privacy-Protected Virtual Writing

In this section, we present a real-world application of RoFin for privacy-protected virtual writing, named *Xamera*. We detail its implementation through three key components: (1) cross-frame video processing with optimized trajectory generation, (2) CNN-based alphanumeric recognition, and (3) rolling shutter configuration and ML model integration in the Android application.

A. Cross-frame Video Processing

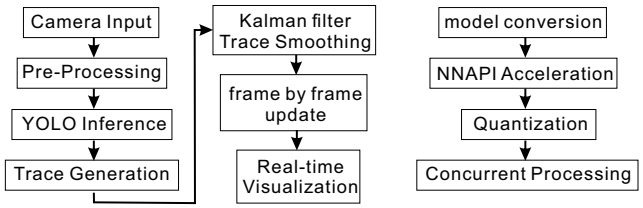


Fig. 16: Cross-frame video processing and optimization flow.

We manage the video processing pipeline in a dedicated class that oversees the entire workflow for the global optimization of virtual writing quality. As illustrated in Figure 16, the workflow includes four stages: (1) **Camera Input and Pre-processing**: The system receives frame-by-frame image data via the Camera2 API. Each frame is duplicated and sent to a pre-processing module, which applies thresholding and noise reduction to the objects of interest (i.e., the bright LED nodes). The processed image is then resized to a fixed resolution of 416×416 to ensure compatibility with the YOLO model for subsequent inference. (2) **Inference and Detection**: The pre-processed frame is converted into a tensor and fed into a customized TFLite YOLO model deployed on the Android device. Non-maximum suppression is then applied to retain only the most confident detection for each class, and the final results (including coordinates and class labels) are scaled back to the original frame dimensions. These results are stored in a queue, which acts as a moving window to reduce memory usage. (3) **Trace Generation and Smoothing**: For each detected object, the center of mass (COM) of its bounding box is calculated to obtain (x, y) coordinates. These coordinates are smoothed using a moving average filter followed by a Kalman filter for enhanced noise reduction. Spline interpolation is then applied to generate additional points between consecutive detections, creating a smooth and continuous trace. The interpolated coordinates are stored in class-specific FIFO queues, which retain only the most recent trace data. (4) **Frame Updates for Real-Time Visualization**: This module runs concurrently with the previous steps, directly receiving the original input frames. It overlays the most recent bounding boxes and spline-generated trace points onto the live video display, providing real-time feedback to the user.

TABLE I: Optimization and Performance.

Technique	Latency Before	Latency After	fps Before	fps After	Growth Factor
Model Conversion (PyTorch → TFLite)	140 ms	80 ms	7 fps	12.5 fps	1.79×
NNAPI Hardware Acceleration	80 ms	30 ms	12.5 fps	33 fps	2.64×
Quantization: FP32 → (FP16 / INT8)	30 ms	22 ms / 17 ms	33 fps	45 fps / 59 fps	1.36× / 1.78×

To achieve real-time performance, we apply four major optimizations to reduce latency on the Google Pixel 8A. The detailed impact of each optimization is as follows and summarized in Table I: (1) **Model Conversion** (PyTorch to TFLite): Converting PyTorch model to TensorFlow Lite reduces inference latency from 140 ms (≈7 fps) to 80 ms (≈12.5 fps), a 1.79× improvement. (2) **NNAPI Hardware Acceleration**: Neural Networks API (NNAPI) enables deployment

on GPUs or dedicated NPUs/TPUs. With NNAPI, latency further decreases from 80 ms to 30 ms (≈12.5 fps to 33 fps), a 2.64× improvement. (3) **Quantization**: Quantization reduces model size by using lower-precision data formats. CPUs typically use FP32, while GPUs are optimized for FP16 to reduce memory usage and increase computational speed. TPUs and NPUs often use INT8, enabling even faster and more efficient computation. In our benchmarks, quantizing from FP32 to FP16 and then to INT8 further reduces latency—from 30 ms to 22 ms and finally to 17 ms (approximately 33 fps to 45 fps to 59 fps). (4) **Concurrency**: Running the inference and frame update modules on separate threads minimizes bottlenecks. Frame updates take under 10 ms, while processing averages 30 ms. This concurrency results in a real-time output of approximately 33 fps.

B. Letter and Digit Recognition for Virtual Writing

After identifying the user and optimizing their traces, we apply CNN models for the recognition of virtual writing.

1) Letter Recognition:

Our letter recognition model classifies uppercase letters using a three-layer convolutional architecture (32, 64, and 128 filters with 3×3 kernels and ReLU activation), as shown in Figure 17 (a). Each convolutional layer incorporates batch normalization and max-pooling for stable training and dimensionality reduction. The network concludes with a 256-unit dense layer (with dropout regularization) and a 26-unit output layer corresponding to the 26 letters of the English alphabet. Inputs are standardized 28×28 grayscale images, with data augmentation applied to enhance model robustness. For training, we combined the Chars74K dataset (1,430 images) with our proprietary RoFin-collected dataset (520 images), followed by application-specific fine-tuning. The Adam optimizer was used with one-cycle learning rate scheduling.

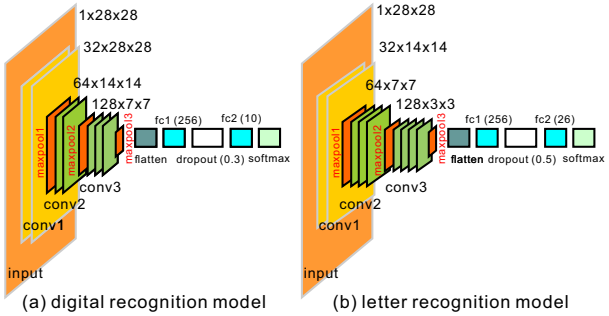


Fig. 17: Applied model architecture for letter/digit recognition.

2) Digit Recognition:

Our digit recognition model adopts a CNN architecture similar to the letter recognition model, as shown in Figure 17 (b). It consists of three convolutional layers with 32, 64, and 128 filters, respectively. The first two layers use a kernel size of 5×5 with a stride of 1 and padding of 2, while the third layer uses a smaller 3×3 kernel with padding of 1. Each convolutional layer is followed by a ReLU activation function and max-pooling. The output is then flattened and passed through two fully connected layers: the first has 256 units with ReLU activation, and the final output layer has 10 units (corresponding to digits 0–9). A dropout layer with a rate of 0.3 is incorporated after the first fully connected layer to mitigate overfitting. To enhance generalization, we applied data augmentation during training. The model was trained on three datasets: MNIST (60,000 images), DIDA (10,000 images), and our RoFin-collected dataset (620 images), as shown in Figure 18. After training on the combined dataset, the model was fine-tuned for RoFin to improve application-specific performance. The Adam optimizer was used with the StepLR scheduler, which reduces the learning rate every 20 epochs to ensure efficient training. Both the letter and digit models utilize consistent CNN architectures and optimization strategies, thereby ensuring robust performance in real-world applications.

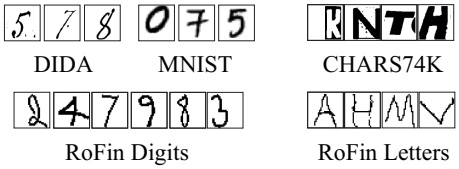


Fig. 18: Training dataset examples for letters and digits.

C. App Development and Model Integration

As one of key use case of RoFin, we developed Xamera app, which seamlessly integrates the aforementioned video processing, YOLO-based object detection, trace smoothing and generation, CNN-based letter/digit recognition, and AR-based visualization modules.

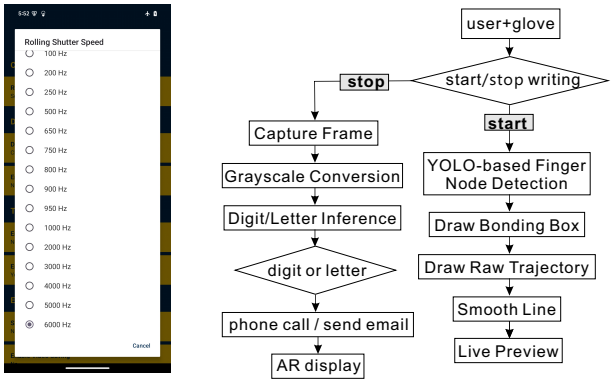


Fig. 19: Shutter configuration in Xamera and its architecture.

We utilize the Camera2 API to provide precise control over manual exposure and hardware settings. In Xamera, a user-configurable shutter speed is offered, as shown in Figure 19 (a). The CameraHelper module then retrieves the user-selected shutter speed and verifies its compatibility with the device's manual exposure range. If compatible, the exposure is clamped, and a safe ISO value is set in full manual mode; otherwise, auto-exposure is enabled. The workflow of the Xamera app is detailed below and illustrated in Figure 19 (b). The Xamera app encompasses three core functional modules: Mode Selection and Writing Controls, Real-time Tracking, and Augmented Reality Integration. For Mode Selection and Writing Controls, users first choose between digit or letter recognition modes to determine the corresponding inference model for processing the captured path; when stopping writing, the app verifies input validity and prompts users to either dial a number or compose an email before launching ARCore for 3D result display. Real-time Tracking starts with initialization via video frame capture through the Camera2 API (introduced in the video processing pipeline VII-A) alongside trace visualization; when tracking stops, the final trace is retrieved from the pipeline and transmitted to the selected recognition model for letter/digit inference. For Augmented Reality Integration, ARCore automatically activates upon completion of the writing session, presenting both a 3D view of the writing trace and a 3D display of the recognized characters.

VIII. System Implementation

A. RoFin Gloves

We implement three wearable RoFin gloves for experiments as shown in Figure 20. The main components in one pair of RoFin gloves are shown in Table II: lightweight insulated breathable gloves, 2 Arduino Nano MCU, 12 green LEDs wrapped with 12 green plastic balls (ϕ = 19mm), and a 9V li-ion battery for power-supply. The spheres we used are industrially produced with the same size and standard sphere shape, thus there is no influence caused by the directivity of the LED and unbalanced brightness issue. The total weight of one pair of RoFin glove is **132g** (including two batteries' weight of 60g) while the total price is only **26.3**$.

TABLE II: Components in one pair of RoFin gloves.

Component	Price (USD)	Details
insulated gloves	0.6 x 2 = 1.2	for each: 24cm x 15cm, 18g
Arduino Nano	10 x 2 = 20	ATmega328P, 5V, 16M
LED	0.02 x 12 = 0.24	5mm, green, 20000mcd, 20mA
plastic cover	0.08 x 12 = 0.96	19mm, green, lightweight
battery	2 x 2 = 4	2 rechargeable batteries 7x2 = 14$
Total price	26.3	mass produced, cheaper the price

B. RoFin Reader

Numerous commercial smart devices are widely available and reasonably priced that can be used as our RoFin reader including smartphones, drone cameras, and underwater cameras. We use commercial smartphones with additional lens such as iPhone 7, VIVO Y71A, and Samsung s20 for experiments, as shown in Figure 20 (b) and (c). The camera parameters including shutter rate, resolution and so on can be configured via camera apps on smartphones (e.g., Protake app).

IX. Performance Evaluation

We evaluate the RoFin's performance in four folds. **(1) label identification** with different ambient light settings, distances, cameras. **(2) inside-frame tracking performance** in contrast to vision-based method. **(3) hand reconstruction performance** with Leap Motion as the benchmark. **(4) virtual writing performance** with our developed Xamera app. Then we also discuss about RoFin's use cases and other concerns such as privacy and power consumption.

A. Robust Label Parsing

(1) Impact of Ambient Light. We use trained YOLO model to predict labels in captured images by VIVO-Y71A in 3 different ambient light settings [low, medium, strong] at the same distance 0.5m of Hand #1. As shown in Figure 21 (a), the label parsing achieves the best accuracy under the strong ambient light at 0.94 and the average accuracy of 0.91. These results demonstrate RoFin's label parsing works robustly under varied ambient light even in the darkness and outperforms than vision-based approaches, which can not work in the darkness and lack of identification ability.

(2) Impact of Sensing Distance. We predict labels in captured images via VIVO-Y71A in 3 distance settings [0.5m, 1.5m, 2.5m] under strong ambient light of Hand #1. The average accuracy of label parsing is shown in Figure 21 (b). The accuracy of label parsing drops slowly with increased distance. RoFin achieves the best accuracy of 0.93 at 0.5m and 0.77 at 2.5m and an average accuracy of 85% label parsing. Although the 15% label parsing error within 2.5m may cause $6\times15\% = 0.9$ key point label parsing error out of a hand, we infer its label based on other parsed labels and improve label parsing accuracy by training with larger dataset. It demonstrates RoFin works robustly under varied distance and outperforms than vision-based approaches (i.e., 1m).

(3) Impact of Different Hands. We also evaluate the label parsing performance of six labels from different hands captured via VIVO-Y71A at 0.5m under strong ambient light. As shown in Figure 21 (c), The hand #1 and #3 achieve high prediction accuracy more than 0.96 while the hand #2 achieves the lowest accuracy of 0.77. The reason is the F6-F10 from hand #2 have more confused rolling patterns than hand #1 and #2. Even though, the average label parsing accuracy still achieves 0.91, which demonstrates the effectiveness of our optical labeling and parsing scheme.

(4) Impact of Different Labels. We also present the confusion matrix of the trained label parsing model for 18 different classes of 3 hands under the settings: VIVO-Y71A, 0.5m and strong ambient light in Figure 21 (d). These 18 classes are finger #1 to finger #15 (F1-F15) and the wrist point #1 to the wrist point #3 (W1-W3). It shows the labels from hand #2 are easier to be identified as other labels than hand #1 and #3, which is consistent with the results in Figure 21 (c). It also shows that the rolling pattern of W2 [CP, On, Off, Off, On, Off] is confused by F6 [CP, Off, Off, On, On, Off]. That is because the reversed rolling patterns of F6 [Off, On, On, Off, Off, CP] (i.e, [..Off, On, On, Off, Off, CP, Off, On, On, Off, Off, CP..]) has the high similarity with the W2 when the amplitude of CP symbol is similar to On symbol.

(5) Impact of Different Cameras. We use the trained model to parse the labels captured by different cameras of commercial smartphones [iPhone 7, VIVO Y71A, and Samsung s20] to measure their label parsing latency performance of hand #1 under strong ambient light at 0.5m with the same resolution of 640×640. As shown in Figure 21 (e), iPhone captured images can be parsed with the shortest time while the average parsing latency of these different cameras are about 12ms (83Hz), and less than the 16.7ms (60 Hz), which demonstrates RoFin achieves the real-time label parsing.

B. Enhanced X/Y Tracking and Depth Parsing

1) Inside-frame X/Y Tracking Performance:

Setup. We bond one fingertips of the RoFin glove with a pen (blue marker) and draw on the transparent plastic paper hanging parallel to the camera's image plane, as shown in Figure 22 (a). We also set two cameras at the fixed distance 0.5m when the user is drawing. One camera follows the traditional vision based approach which captures the video as usual with 60 fps frame rate while the other camera (RoFin reader) captures the video of the rolling patterns with the same 60 fps frame rate but with high rolling shutter rate (**8 kHz**). Thus we track 3 traces of user's drawing at the same time: (1) ground truth on the plastic paper, (2) vision approach tracked trace, (3) RoFin tracked trace.

X/Y Tracking Enhancement. We ask a user to draw 3 different letters: (1) M with more straight lines, (2) C with curve, (3) a rotated α with more complex curve with two writing speed: (1) normal speed, and (2) faster speed. We process the captured videos with 6 above setting combinations via PotPlayer software to generate frames for

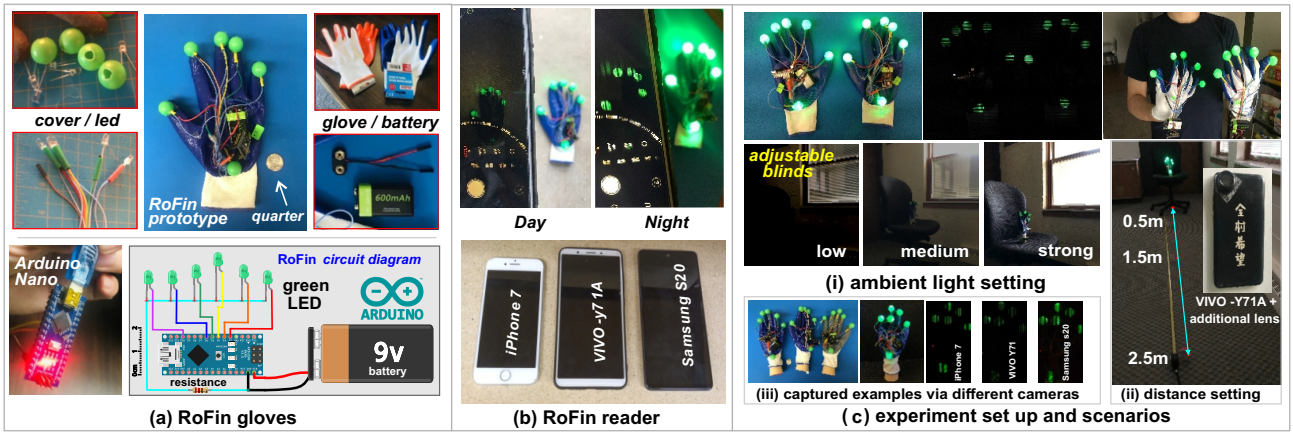


Fig. 20: System implementation: RoFin gloves (prototype & circuit diagram), RoFin reader (commercial cameras) and experiment scenarios (varied ambient light strength and distances from 0.5m to 2.5m).

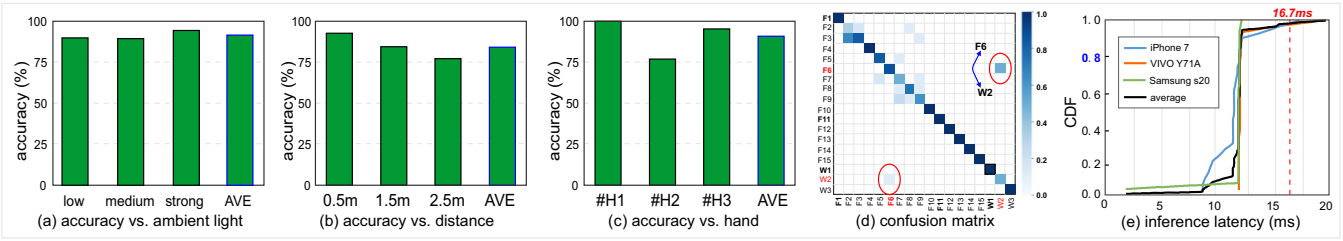


Fig. 21: Label parsing accuracy performance in varied settings and latency evaluation.

measurement. As shown in Figure 22 (b) and (c), the RoFin tracked 4 times of location points for the same letter, which significantly enhances the granularity of tracking trace in compared to vision-based tracking. Besides, RoFin achieves more accurate trace tracking than vision based method among all three different letters due to its fine-grained inside-frame sampling. We also conduct experiments to figure out the up-sampling ability of RoFin via set higher rolling shutter rate at 12 kHz, 16 kHz and 20 kHz while keep the motion speed the same and high. As shown in Figure 22 (d), with the increased transmission frequency with matched rolling shutter rate, the maximum number of sampled points in one frame increases from 7 sampled points to 8, 10, and 12 points separately because of the decreased width of a rolling strip.

2) Depth Z Estimation Performance:

Setup. We set a wooden hand model worn RoFin glove at the desk, as shown in Figure 22 (d). The fingertips are separated with different distances to the camera image plane (XY plane). The hand model keeps the same pose but with 3 different orientations to the camera. We also set camera with 3 different rotations to capture the RoFin glove. Then we measure the distance between the fingertips' projected points on the desk to the camera plane as the Z ground truth. We measure all 6 key points in evaluation under 9 different setting combination of orientation and rotation at distance of 0.5m.

Z Estimation Accuracy. As shown in Figure 22 (e), although the error of estimated depth info Z via RoFin varies with the different hand orientation and camera rotation, RoFin achieves the average estimation error of 1.6 cm.

Summary. The experiment results in Section IX-B1 and IX-B2 demonstrates that our low-cost RoFin provides enhanced X/Y tracking with up-sampling ability of 12X of vision based approach and accurate Z estimation for depth parsing.

C. Real-time Hand Pose Reconstruction

We define 10 hand poses as shown in Figure 23 for hand pose reconstruction evaluation. We capture the images of the wooden hand worn the RoFin glove with RoFin reader for different hand poses. Then we run HPR model and evaluate its accuracy and latency with Leap Motion as benchmark.

1) Reconstructing Accuracy:

Impact of Ambient Light. We define the deviation error as the average difference of x,y,z between RoFin with Leap Motion. As shown in Figure 24 (b), the average deviation error of three ambient light settings [low, medium, strong] under 0.5m has the similar distribution and the most deviation error is distributed less than 22 mm. Among three ambient light settings, the medium ambient light achieves the best performance due to the RoFin reader can capture the most clear contours of six key points' spheres.

Impact of Sensing Distance. As shown in Figure 24 (c), the average deviation error of three distances [0.5m, 1.5m, 2.5m] are similar and are mostly distributed in 28 mm. The deviation error of 1.5m achieves the best performance with the average deviation error of 14 mm while the 2.5m setting achieves the largest average deviation error of 19 mm. These results demonstrate our HPR model works well up to 2.5 m while the vision approaches usually work within 1 m and Leap Motion works within 0.5m.

Impact of Different Poses. We also evaluate the reconstructing deviation error of 10 hand poses defined above at 0.5m. As shown in Figure 24 (c), the reconstructed y has the largest deviation error compared with x and z, especially for the hand pose (b), point with index finger. The reason is that the finger planes of the ring finger, the little finger are not exactly as assumed in our simplified HPR model that their finger planes perpendicular to the projected palm plane. Among 10 hand poses, the pose (j) achieves the lowest average deviation error in hand pose reconstructing of 7.6 mm.

Impact of Dynamic Scenarios. Besides the wooden hand based evaluation above, we ask the user to wear RoFin gloves for dynamic scenario evaluation, as shown in Figure 25. The user stands 0.3m far away from RoFin camera and Leap Motion and uses the fixed hand pose shown in the right of Figure 25 with four different motion speeds [static, slow, medium, fast]. As shown in Figure 24 (d), the average deviation error with different motion speeds are similar and less than 4cm. The reason is that the X/Y/Z pair for hand pose reconstruction are based on frame level one-sample parsing instead of multiple-

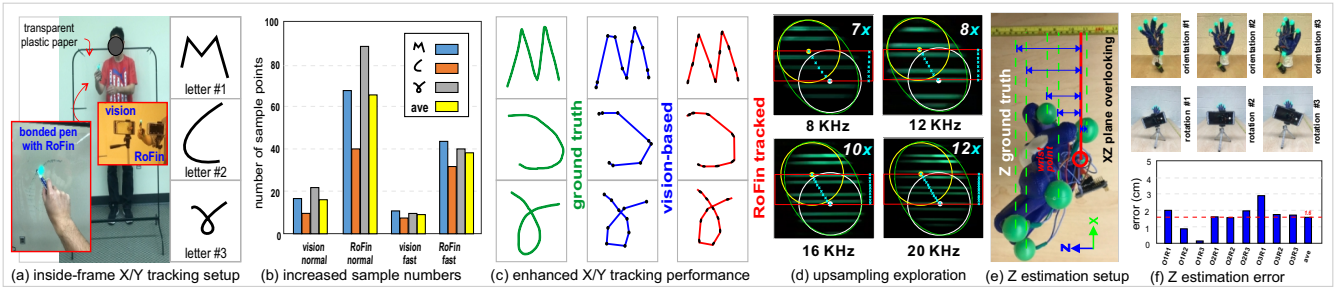


Fig. 22: Enhanced inside-frame tracking with up-sampling exploration, and Z estimation performance.

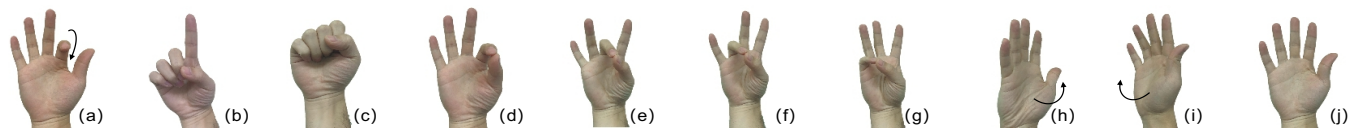
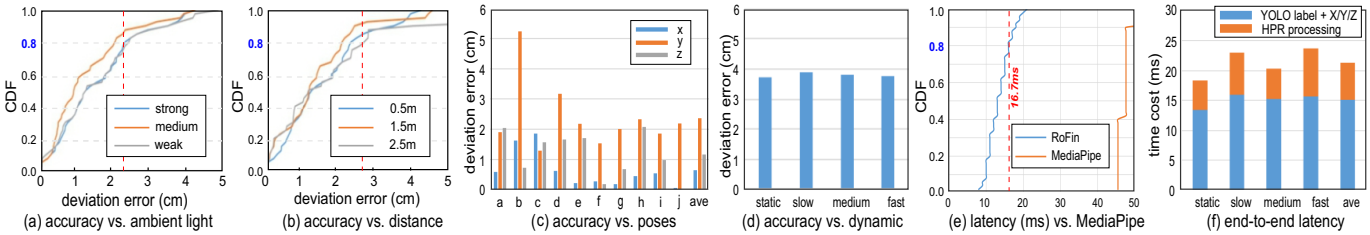


Fig. 23: 10 defined hand poses: (a) bend index finger, (b) point with index finger, (c) close the fist, (d-g) pinch thumb with Index, Middle, Ring, and Little finger, (h) turn palm to the left, (i) turn palm to the right, (j) the palm.



(a) accuracy vs. ambient light (b) accuracy vs. distance (c) accuracy vs. poses (d) accuracy vs. dynamic (e) latency (ms) vs. MediaPipe (f) end-to-end latency

Fig. 24: (a) reconstruction deviation error CDF under different ambient light setting, (b) reconstructing deviation error CD under different distance, (c) deviation error for different hand poses, (d) deviation error under different dynamic setting, (e) reconstructing latency comparison between RoFin and MediaPipe, and (f) end to end latency performance of RoFin.

sampled X/Y pairs for finger tracking.

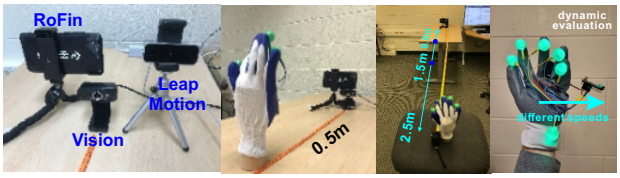


Fig. 25: Experiment scenarios.

2) Reconstructing Latency.: As for hand pose reconstructing, the main advantage of RoFin compared with vision-based approaches is its less tracked key points and flexible and long sensing distance. We evaluate the hand pose reconstructing latency and make comparison with the vision based approach Media Pipe ran on the same platform: Thinkpad T480 with Intel(R) Core(TM) i7-8650U CPU for different hand poses under the same 0.5m distance and strong ambient light setting.

As shown in Figure 24 (e), the latency of the RoFin HPR model is distributed less than 21 ms with the average latency of 13.8 ms (72Hz), which is less than 16.7 ms (60Hz). The vision based Media Pipe achieves 47.5 ms latency in average. Although the finger label parsing requires about 12ms for each image frame, the label parsing module and the HPR module can still run in pipe-line manner to achieve the real-time processing. These results demonstrate that our HPR model can achieve real-time hand pose reconstructing due to its only tracking 6 key points with simplified HPR model.

We also measure the end to end latency for hand pose reconstruction including (1) finger identification, (2) YOLO based X/Y parsing, (3) Z estimation, and (4) HPR processing under 4 different motion speed of the same hand pose. (1)-(3) are processed together and the output is the input of step (4). As shown in Figure 24 (f), there is no significant difference in the end-to-end time cost among different motion speeds.

D. Use Case: RoFin-enabled Virtual Writing

We implement the virtual writing function for RoFin in the Xamera app, as shown in Figure 26 (a). We evaluate the performance of virtual writing, including (1) the cross-frame video processing latency, (2) the trace smoothness, and (3) the letter/digit recognition accuracy under three different office lighting conditions (i.e., with artificial light in daytime, without artificial light in daytime, and in the dark) at a distance of 0.5 meters, as shown in Figure 26 (b).

Cross-frame video processing latency. We use 3,000 video frames to measure the cross-frame video processing latency. As shown in Figure 26 (c), the end-to-end cross-frame video processing in Xamera (including YOLO-based user identification, trace generation, Kalman filter-based trace smoothing, and visualization) has an average latency of 55.2 ms.

Trace smooth rate. We define the smooth ratio for our trace smoothing as the $S_{ratio} = \frac{d_o - d_s}{d_o}$, where d_o and d_s denote the pixel gap between the highest and lowest points of the original trace and the smoothed trace, respectively. We use 60 frames to measure the smoothing performance. As shown in Figure 26 (d), 23.3% of the frames exhibit a smooth ratio in the range [0, 0.1], 30% in [0.1, 0.2], 25% in [0.2, 0.3], 10% in [0.3, 0.4], and 8.3% in [0.4, 0.5]. The average smooth ratio of 0.19 confirms effective jitter mitigation.

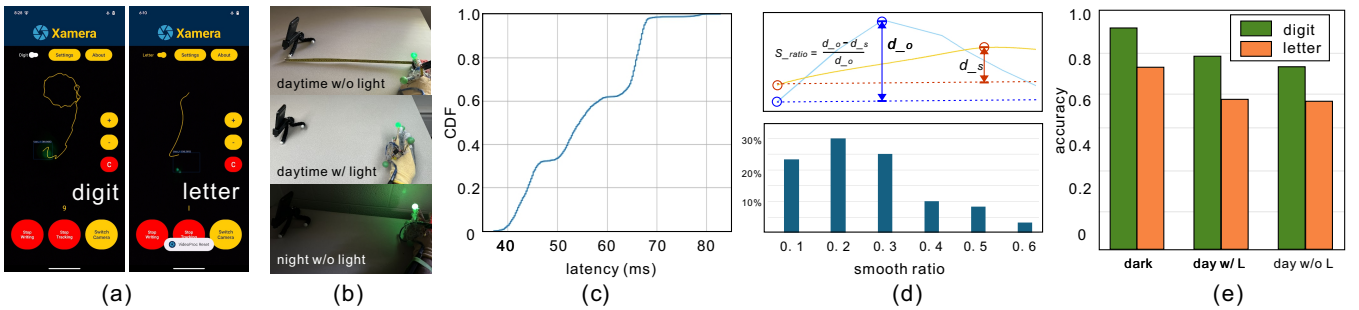
Letter/Digit Recognition Accuracy. As shown in Figure 26 (e), digit recognition consistently achieves higher accuracy than letter recognition across all three environmental settings. The average digit recognition accuracy is 0.85, compared to 0.71 for letter recognition. No significant accuracy difference is observed between daytime indoor conditions with and without artificial lighting for both tasks.

E. Other Use Cases

Multi-user interaction for AR/VR/MR. As demonstrated in Section IX-B1, RoFin can track inside-frame X/Y location samples at rolling shutter rate and thus provide the ability of fine-grained finger tracking, especially for the high-speed motion or small-scale motion. Multiple users can use their fingertips to write or paint virtually at the same time in front of the camera. Thus, RoFin can be used as the user interface with better user experience for AR/VR/MR with privacy protection of users due to they only want the camera to capture the trace instead of the face, as shown in Figure 27 (a).

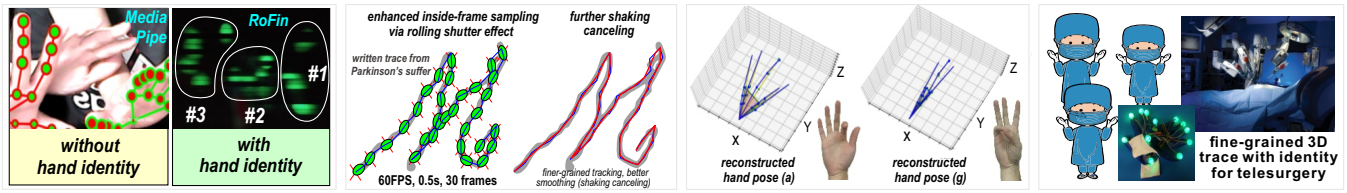
Hand Pose Commands for Video Games/Smart Home. As demonstrated in Section IX-C, RoFin achieves real-time hand pose reconstructing with less computation overhead and high accuracy. Our low-cost RoFin system can be used as the hand pose command input interface for video games, smart home, etc. Figure 27 (c) shows reconstructed hand pose examples via RoFin's HPR model.

Telesurgery Gloves. There are doctors, nurses, and remote operation robots involved in telesurgery. RoFin can distinguish whose



(a) (b) (c) (d) (e)

Fig. 26: Case study of virtual writing. (a) Xamera interface, (b) experiment scenarios, (c) cross-frame video processing latency, (d) trace smooth rate, (e) letter/digit recognition accuracy.



(a) multi-user MR interaction (b) fine-grained trampling tracking (c) hand pose commands (d) telesurgery interfaces

Fig. 27: 4 possible use cases for our low-cost RoFin: (1) multi-user MR interactions with identification and protected privacy, (2) finer-grained tracking of writing of Parkinson's suffer, (3) real-time hand pose commands, and (4) telesurgery interfaces.

TABLE III: Comparison of RoFin with mainstream non-vision based commercial wearable gloves.

Device	Sampling / Update Rate	Latency	Power	Battery Life	Spatial Accuracy	Cost
RoFin (ours)	up to 20 kHz; 720 Hz	< 16.7 ms	avg. 225 mW	~ 24 h	< 22 mm	\$26.3
Manus Metagloves [41]	120 Hz	≤ 7.5 ms	Not publicly disclosed	~ 4 h	mm-level (claimed)	$\sim$ \$5,000
Dexmo Glove [42], [43]	~ 462 Hz internal	Not publicly disclosed	avg. ~ 1 W	~ 5 h	Not publicly disclosed	$\leq$ \$3,000
SenseGlove Nova 2 [44]	~ 60 Hz	10–29 ms	Not publicly disclosed	~ 3 h	Not publicly disclosed	\$5,999

hand it is using fine-grained real-time finger traces and then send the 3D traces to the remote surgical robots for operations, as shown in Figure 27 (d). Our goal is to sense fine-grained 3D traces without sensing delay, rather than directly performing surgery using RoFin gloves; thus, aspects requiring future optimization will not be issues.

X. Discussion

Power Consumption and Safety. Our RoFin gloves are made of electric insulation rubber gloves, and the voltage at the LED node side is less than 3V, ensuring the safety of users who wear gloves. The current through one RoFin glove's circuit is **75** mA, and the power consumption is **225** mW. Based on our 600mAh and 9V li-ion battery, one RoFin glove can work for approximately 5.4 Wh / 225 mW = **24** hours before needing to be recharged.

Non-vision based Solutions. There are two types of Non-vision based solutions: (1) on-body sensor based approaches[45], [46], [47], and (2) hand-free approaches[48], [12], [13], [1]. Compared with our RoFin, these approaches are limited in: (1) requirement of specific/expensive sensors and devices instead of commercial LED nodes, such as mmWave chips, FBG sensors, (2) sensing distances within the near hand area (i.e., within 0.5m), (3) lack of finger/hand identification and can not serve multiple users with identification, (4) wearable/IMU solutions can sample at a high frequency at the user side, but they must incur data-send-out overhead and latency for centralized control (e.g., in a situation where multiple users are actively engaged), whereas RoFin obtains these data directly at non-user sides without incurring data-send-out overhead.

Comparison with Commercial Products. (1) To precept hands' morphological variation frame by frame, Leap Motion's [24] uses infrared cameras with illuminating LEDs. (2) XSens[11] performs 3D motion capturing made possible by specialized and pricey tiny MEMS inertial sensors. For example, the MOVELLA DOT SENSOR costs \$132. (3) Instead of using hand posture reconstruction, Luxapose [25] utilizes many fixed LED landmarks on the ceiling to work with a rolling camera for indoor localization. Compared with mainstream wearable glove systems, RoFin offers clear advantages in cost, sampling frequency, power efficiency, and real-time performance. Manus Metagloves operate at a 120 Hz sensor rate with ≤ 7.5 ms latency and about four hours of battery life [41]. Dexmo gloves provide an internal update rate of approximately 462 Hz but require higher power consumption with roughly 1 W average and about five hours of battery life [42], [43]. SenseGlove Nova 2 focuses on haptics and typically reports a 60 Hz tracking rate with 10–29 ms latency [44]. By contrast, RoFin achieves a prototype cost of 26.3 USD, supports up to 20 kHz sensing and 720 Hz positional updates (i.e., 60 fps frame refresh rate multiplied by 12 location samples per frame), operates at only 225 mW per glove, and reaches ≤ 16.7 ms latency with ≤ 22 mm spatial accuracy. These results indicate that RoFin provides substantially higher frequency tracking, lower power consumption, and significantly lower cost than commercially available glove systems. The detailed comparisons are listed in Table III.

Privacy Leakage. Pure-vision approaches (i.e., eyes, MediaPipe, Leap Motion integrated with camera) can cause privacy leakage [24], [49]. As shown in Figure 28, the user performs hand pose commands while holding a bank card. Leap Motion and MediaPipe leak sensitive and private information (e.g., CVV numbers, user's face, hands, background), posing risks of property loss. In contrast, RoFin leverages the rolling shutter's high frequency to only perceive active, intense light (i.e., LED nodes on the glove) and fundamentally exclude passively reflected light from the user's face, hands, and background. Per Privacy Impact Assessment (PIA) criteria, this hardware-level perception design eliminates collection of sensitive visual data (e.g., facial features, environmental details), avoiding privacy leakage risks associated with pure-vision interaction methods [50].

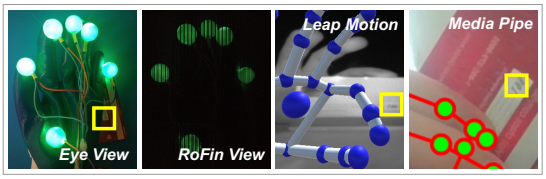


Fig. 28: Sensitive data leakage of vision-SOTA.

Glove Size Considerations for User Experience and Generality. The RoFin prototype uses a universal one-size glove similar to commercial work gloves. Anthropometric data and manufacturer sizing charts show that typical XS–XL ranges (≈17–24 cm palm width) already cover most adult users [51], [52]. Because RoFin relies on rolling-shutter LED pattern recognition and tracks only the fingertips and wrist nodes rather than all joints or precise geometric conformity, minor fit variation has limited impact on tracking. Prior studies on elastic textile wearables further confirm that stretchable fabrics maintain functional performance across diverse hand shapes [53], [54]. For future versions, we can adopt a multi-size glove strategy (ES/S/M/L), following industry practice as exemplified by MANUS' Metagloves Pro [55].

Limitations of RoFin and Future Evolution Pathways. Compared with hands-free approaches (e.g., vision-based methods), the current RoFin prototype requires users to wear gloves attached with plastic spheres, wires, and a battery, but this limitation can be effectively addressed through targeted optimizations including ergonomic design, textile integration, energy harvesting, and passive labeling: future iterations will adopt an ergonomically designed lightweight elastic fabric glove with embedded wires (aligning with the trend of textile-based wearable miniaturization [56]), paired with a miniaturized control core (Seeeduino Xiao [57], $20 \times 17.5 \times 3.5$ mm) and 3.7V micro button battery to reduce total weight and size, while the plastic spheremounted LEDs could be replaced with modular, passivelabellike nailart patches powered by micro energy

harvesting modules (e.g., piezoelectric [58]) that capture energy from finger movement. This eliminates the need for external batteries, and the patches are pre-programmed with light patterns to replicate RoFin's high-frequency encoding. The technology could potentially interoperate with existing commercial smart-gloves (e.g. Manus [55], Dexmo [59], SenseGlove [60]), which are documented in the literature and commercially available [61]. All of these gloves have been widely adopted in XR interaction, industrial training, and surgical simulation to minimize user adaptation barriers. Notably, while hands-free vision-based methods offer convenience, they sacrifice high-frequency tracking accuracy ($\leq$60Hz) and privacy as discussed above, whereas our optimized wearable design prioritizes 720 Hz tracking and hardware-level privacy. This is critical for core scenarios including industrial precision assembly, VR surgical simulation, and high-frequency e-sports interaction, where accuracy and security outweigh hands-free convenience. This ensures the current prototype and future iterations are fully aligned with these practical use cases.

XI. Conclusion

In this paper, we first exploit the 2D temporal-spatial rolling to construct 3D hand pose. We address technical challenges in RoFin design and implementation, e.g, fingertips active optical labeling, fine-grained 3D information parsing of rolling fingertips, and lightweight 20-joints 3D hand pose reconstructing via 6 tracked key points. Then we undertake studies using RoFin gloves in a variety of circumstances. The results demonstrate RoFin can robustly identify fingers, parse fine-grained 3D info, and achieve real-time hand pose reconstruction. We also developed Xamera as a RoFin reader for real-world applications in privacy-protected air writing with digit/letter recognition. Our RoFin is a low-cost but effective solution for human computer interactions with promising use cases.

References

[1] T. Li, X. Xiong, Y. Xie, G. Hito, X.-D. Yang, and X. Zhou, "Reconstructing hand poses using visible light," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, pp. 1–20, 2017.

[2] J. Gong, Y. Zhang, X. Zhou, and X.-D. Yang, "Pyro: Thumb-tip gesture recognition using pyroelectric infrared sensing," in *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, 2017, pp. 553–563.

[3] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, "Mediapipe hands: On-device real-time hand tracking," *arXiv preprint arXiv:2006.10214*, 2020.

[4] R. Zhao, D. Wang, Q. Zhang, X. Jin, and K. Liu, "Smartphone-based handwritten signature verification using acoustic signals," *Proceedings of the ACM on Human-Computer Interaction*, vol. 5, no. ISS, pp. 1–26, 2021.

[5] X. Zhang, G. Klevering, and L. Xiao, "Exploring rolling shutter effect for motion tracking with objective identification," in *Proceedings of the Twentieth ACM Conference on Embedded Networked Sensor Systems*, 2022, pp. 816–817.

[6] X. Zhang, H. Guo, J. Mariani, and L. Xiao, "U-star: An underwater navigation system based on passive 3d optical identification tags," in *The 28th Annual International Conference on Mobile Computing and Networking*, 2022.

[7] X. Zhang, G. Klevering, X. Lei, Y. Hu, L. Xiao, and T. Guanhua, "The security in optical wireless communication: A survey," *ACM Computing Surveys (CSUR)*, 2023.

[8] X. Zhang and L. Xiao, "Rainbowrow: Fast optical camera communication," in *2020 IEEE 28th International Conference on Network Protocols (ICNP)*. IEEE, 2020, pp. 1–6.

[9] X. Zhang, G. Klevering, and L. Xiao, "Posefly: On-site pose parsing of swarming drones via 4-in-1 optical camera communication," in *2023 IEEE 24th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2023, pp. 1–10.

[10] X. Zhang, G. Klevering, K. Wijewardena, and L. Xiao, "Demo: Integrated on-site localization and optical camera communication for drones," in *2023 IEEE 24th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2023, pp. 1–3.

[11] (2023) XSens. [Online]. Available: https://www.movella.com/

[12] J. Lien, N. Gillian, M. E. Karagozler, P. Amihood, C. Schwesig, E. Olson, H. Raja, and I. Poupyrev, "Soli: Ubiquitous gesture sensing with millimeter wave radar," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–19, 2016.

[13] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota, "Fingerio: Using active sonar for fine-grained finger tracking," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016, pp. 1515–1525.

[14] S. Sur, I. Pefkianakis, X. Zhang, and K.-H. Kim, "Towards scalable and ubiquitous millimeter-wave wireless networks," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 2018, pp. 257–271.

[15] A. Galisteo, Q. Wang, A. Deshpande, M. Zuniga, and D. Giustiniano, "Follow that light: Leveraging leds for relative two-dimensional localization," in *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*, 2017, pp. 187–198.

[16] Y. Yang, J. Hao, and J. Luo, "Ceilingtalk: Lightweight indoor broadcast through led-camera communication," *IEEE Transactions on Mobile Computing*, vol. 16, no. 12, pp. 3308–3319, 2017.

[17] X. Zhang and L. Xiao, "Lighting extra data via owc dimming," in *Proceedings of the Student Workshop*, 2020, pp. 29–30.

[18] X. Zhang, J. Mariani, L. Xiao, and M. W. Mutka, "Lifod: Lighting extra data via fine-grained owc dimming," in *2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2022, pp. 73–81.

[19] X. Zhang, G. Klevering, J. Mariani, L. Xiao, and M. W. Mutka, "Boosting optical camera communication via 2d rolling blocks," *Proceedings of IEEE/ACM International Symposium on Quality of Service*, 2023.

[20] X. Zhang, G. Klevering, J. Wang, L. Xiao, and T. Li, "Rofin: 3d hand pose reconstructing via 2d rolling fingertips," *Proceedings of 21st ACM International Conference on Mobile Systems, Applications, and Services, conditionally accepted*, 2023.

[21] Y. Yang and J. Luo, "Boosting the throughput of led-camera vlc via composite light emission," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 315–323.

[22] ——, "Composite amplitude-shift keying for effective led-camera vlc," *IEEE Transactions on Mobile Computing*, vol. 19, no. 03, pp. 528–539, 2020.

[23] X. Zhang, G. Klevering, J. Mariani, L. Xiao, and M. W. Mutka, "Boosting optical camera communication via 2d rolling blocks," in *2023 IEEE/ACM 31st International Symposium on Quality of Service (IWQoS)*. IEEE, 2023, pp. 1–4.

[24] (2023) Leap Motion Tutorial. [Online]. Available: https://en.wikipedia.org/wiki/Leap_Motion

[25] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta, "Luxapose: Indoor positioning with mobile phones and visible light," in *Proceedings of the 20th annual international conference on Mobile computing and networking*, 2014, pp. 447–458.

[26] R. Wang, S. Paris, and J. Popović, "6d hands: markerless hand-tracking for computer aided design," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 2011, pp. 549–558.

[27] Y. Cai, L. Ge, J. Cai, and J. Yuan, "Weakly-supervised 3d hand pose estimation from monocular rgb images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 666–682.

[28] R. Li, Z. Liu, and J. Tan, "A survey on 3d hand pose estimation: Cameras, methods, and datasets," *Pattern Recognition*, vol. 93, pp. 251–272, 2019.

[29] A. Tagliasacchi, M. Schröder, A. Tkach, S. Bouaziz, M. Botsch, and M. Pauly, "Robust articulated-icp for real-time hand tracking," in *Computer graphics forum*, vol. 34, no. 5. Wiley Online Library, 2015, pp. 101–114.

[30] R. Rastgoo, K. Kiani, and S. Escalera, "Sign language recognition: A deep survey," *Expert Systems with Applications*, vol. 164, p. 113794, 2021.

[31] M. A. Thenganatt and E. D. Louis, "Distinguishing essential tremor from parkinson's disease: bedside tests and laboratory evaluations," *Expert review of neurotherapeutics*, vol. 12, no. 6, pp. 687–696, 2012.

[32] M. Cui, Q. Wang, and J. Xiong, "Breaking the limitations of visible light communication through its side channel," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, 2020, pp. 232–244.

[33] S. Zhu, C. Zhang, and X. Zhang, "Lishield: Create a capture-resistant environment against photographing," in *Proceedings of the 9th ACM Workshop on Wireless of the Students, by the Students, and for the Students*, 2017, pp. 23–23.

[34] Y. Wu, P. Wang, K. Xu, L. Feng, and C. Xu, "Turboboosting visible light backscatter communication," in *Proceedings of the Annual conference*

of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication, 2020, pp. 186–197.

[35] "Ieee standard for local and metropolitan area networks–part 15.7: Short-range optical wireless communications," *IEEE Std 802.15.7-2018 (Revision of IEEE Std 802.15.7-2011)*, pp. 1–407, April 2019.

[36] M. Rezaei, R. Rastgoo, and V. Athitsos, "Trihorn-net: A model for accurate depth-based 3d hand pose estimation," *Expert Systems with Applications*, vol. 223, p. 119922, 2023.

[37] G. Pavlakos, D. Shan, I. Radosavovic, A. Kanazawa, D. Fouhey, and J. Malik, "Reconstructing hands in 3d with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 9826–9836.

[38] W. Cheng, H. Tang, L. Van Gool, and J. H. Ko, "Handdiff: 3d hand pose estimation with diffusion on image-point cloud," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 2274–2284.

[39] S. Yu, Y. Wang, L. Chen, X. Zhang, and J. Li, "3d hand pose and mesh estimation via a generic topology-aware transformer model," *Frontiers in Neurorobotics*, vol. 18, p. 1395652, 2024.

[40] A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir, "Inverse kinematics techniques in computer graphics: A survey," in *Computer graphics forum*, vol. 37, no. 6. Wiley Online Library, 2018, pp. 35–58.

[41] Manus, "Manus metagloves pro product specifications," 2023, online product documentation.

[42] J. Mulder, "Evaluation of dexmo glove update rate using ros interface," 2023, bachelor Thesis.

[43] D. Robotics, "Dexmo glove technical and power specifications," 2022, commercial product documentation.

[44] SenseGlove, "Senseglove nova and nova 2 technical documentation," 2023, developer documentation.

[45] W. Chen, C. Yu, C. Tu, Z. Lyu, J. Tang, S. Ou, Y. Fu, and Z. Xue, "A survey on hand pose estimation with wearable sensors and computer-vision-based methods," *Sensors*, vol. 20, no. 4, p. 1074, 2020.

[46] Y. Lee, M. Kim, Y. Lee, J. Kwon, Y.-L. Park, and D. Lee, "Wearable finger tracking and cutaneous haptic interface with soft sensors for multi-fingered virtual manipulation," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 1, pp. 67–77, 2018.

[47] J. S. Kim, B. K. Kim, M. Jang, K. Kang, D. E. Kim, B.-K. Ju, and J. Kim, "Wearable hand module and real-time tracking algorithms for measuring finger joint angles of different hand sizes with high accuracy using fbg strain sensor," *Sensors*, vol. 20, no. 7, p. 1921, 2020.

[48] H. Xu, D. Iwai, S. Hiura, and K. Sato, "User interface by virtual shadow projection," in *2006 SICE-ICASE International Joint Conference*. IEEE, 2006, pp. 4814–4817.

[49] (2022) mediapipe. [Online]. Available: https://github.com/google-ai-edge/mediapipe

[50] E. McCallister, T. Grance, and K. Scarfone, "Guide to protecting the confidentiality of personally identifiable information (pii)," National Institute of Standards and Technology, Gaithersburg, MD, Tech. Rep., 2010. [Online]. Available: https://csrc.nist.gov/publications/detail/sp/800-122/final

[51] NC State Ergonomics Center, "Anthropometry summary tables (hand breadth, length percentiles)," https://multisite.eos.ncsu.edu/www-ergocenter-ncsu-edu/wp-content/uploads/sites/18/2016/06/Anthropometric-Detailed-Data-Tables.pdf.

[52] "Glove size in cm," https://www.mechanix.com/on/demandware.static/-/Library-Sites-MechanixSharedLibrary/default/dwa6059da6/pdf/Glove%20Size_IN_CM.pdf, Mechanix Wear.

[53] Y. Zhang, J. Zhou, Y. Zhang, D. Zhang, K. T. Yong, and J. Xiong, "Elastic fibers/fabrics for wearables and bioelectronics," *Advanced Science*, vol. 9, no. 35, p. 2203808, 2022.

[54] A. Talukder, S. Choudhury, and et al., "Elastic textile-based wearable modulation of musculoskeletal load: a comprehensive review," *Wearable Technologies*, vol. 3, no. e15, 2025.

[55] MANUS Technology Group, "Manus metagloves pro – high-precision, low-latency data gloves," https://www.manus-meta.com/products/metagloves-pro.

[56] Y. Li, C. Zheng, S. Liu, L. Huang, T. Fang, J. X. Li, F. Xu, and F. Li, "Smart glove integrated with tunable mwnts/pdms fibers made of a one-step extrusion method for finger dexterity, gesture, and temperature recognition," *ACS Applied Materials & Interfaces*, vol. 12, no. 21, pp. 23 764–23 773, 2020.

[57] Seeed Studio, "Seeeduino xiao — getting started / hardware overview," https://wiki.seeedstudio.com/Seeeduino-XIAO/, accessed: 2025-12-03.

[58] I. Sobianin, S. D. Psoma, and A. Tourlidakis, "A 3d-printed piezoelectric microdevice for human energy harvesting for wearable biosensors," *Micromachines*, vol. 15, no. 1, p. 118, 2024.

[59] D. Robotics, "Dexmo – force feedback haptic gloves," https://www.dextarobotics.com.

[60] SenseGlove, "Senseglove nova – haptic vr glove with force and vibrotactile feedback," https://www.senseglove.com/.

[61] M. Caeiro-Rodríguez, I. Otero-González, F. A. Mikic-Fonte, and M. Llamas-Nistal, "A systematic review of commercial smart gloves: Current status and applications," *Sensors*, vol. 21, no. 8, p. 2667, 2021.