Reconstructing Hand Poses Using Visible Light

TIANXING LI*, XI XIONG*, YIFEI XIE, GEORGE HITO, XING-DONG YANG, and XIA ZHOU,
Dartmouth College

Free-hand gestural input is essential for emerging user interactions. We present Aili, a table lamp reconstructing a 3D hand skeleton in real time, requiring neither cameras nor on-body sensing devices. Aili consists of an LED panel in a lampshade and a few low-cost photodiodes embedded in the lamp base. To reconstruct a hand skeleton, Aili combines 2D binary blockage maps from vantage points of different photodiodes, which describe whether a hand blocks light rays from individual LEDs to all photodiodes. Empowering a table lamp with sensing capability, Aili can be seamlessly integrated into the existing environment. Relying on such low-level cues, Aili entails lightweight computation and is inherently privacy-preserving. We build and evaluate an Aili prototype. Results show that Aili's algorithm reconstructs a hand pose within 7.2 ms on average, with 10.2° mean angular deviation and 2.5-mm mean translation deviation in comparison to Leap Motion. We also conduct user studies to examine the privacy issues of Leap Motion and solicit feedback on Aili's privacy protection. We conclude by demonstrating various interaction applications Aili enables.

CCS Concepts: •**Human-centered computing** →**Gestural input;** *Interface design prototyping; Ambient intelligence; Ubiquitous and mobile computing systems and tools;*

Additional Key Words and Phrases: Gestural input, 3D hand reconstruction, visible light sensing

ACM Reference format:
Tianxing Li*, Xi Xiong*, Yifei Xie, George Hito, Xing-Dong Yang, and Xia Zhou. 2017. Reconstructing Hand Poses Using Visible Light. *PACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article 71 (September 2017), 20 pages.
DOI: http://doi.org/10.1145/3130937

1 INTRODUCTION

Recent advances in smart home appliances have drastically enriched user experiences in indoor environments such as homes and offices. However, interacting with smart appliances, either on the appliances themselves or through the use of a smartphone, is still quite cumbersome. As demonstrated by smart TVs [1] and smoke alarms [2], free-hand gestural input has great potential for relieving the interaction burden. It suggests that precise, arbitrary hand gestures may soon become the primary input modality for interacting with smart appliances.

To sense free-hand gestures, existing approaches have examined the use of cameras (e.g., RGB or infrared cameras), on-body sensors (e.g., capacitive sensors, pressure sensors), ambient radio frequency (e.g., Wi-Fi, GSM) signals, and acoustic signals. Most approaches focus on differentiating a small set of pre-defined gestures, thus limiting the range of user input and achieving a coarse sensing granularity. The approaches capable of

(*) T. Li and X. Xiong are co-primary authors of the paper.

This work is supported by the National Science Foundation, under grants CNS-1552924 and CNS-1421528.

Authors' addresses: T. Li and X. Xiong and Y. Xie and G. Hito and X. Yang and X. Zhou, Computer Science Department, Dartmouth College, NH 03755, US..

DOI: http://doi.org/10.1145/3130937

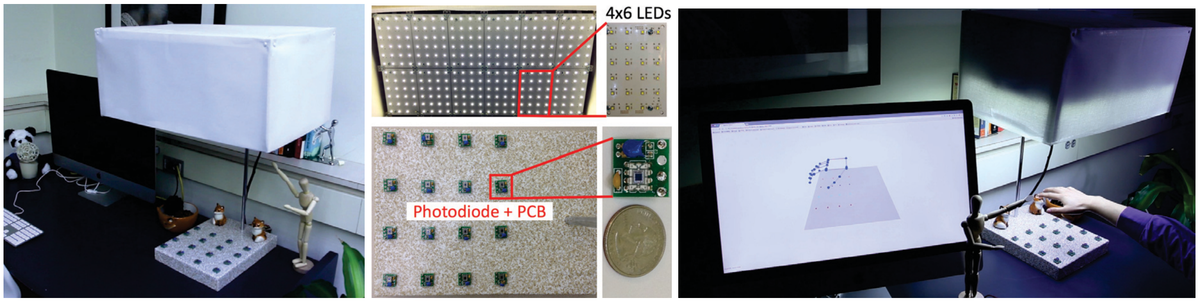


Fig. 1. Aili looks like a regular lamp, but it also reconstructs arbitrary hand poses in real time, with neither cameras nor on-body sensors.

reconstructing arbitrary hand poses commonly rely on cameras, which often entail non-trivial computational overhead in dealing with a large number of gray/RGB-scale pixels in camera images.

In this work, we propose a lightweight alternative approach that reconstructs hand poses purely based on a set of *binary* blockage information sensed by a few low-cost photodiodes, requiring neither cameras nor on-body sensors. Realized as a table lamp, our system Aili consists of a customized LED panel (with arrays of LEDs) in a lampshade, and a few small (the sensing area is 10 mm × 7 mm in size) photodiodes embedded in the lamp base. A user performs free-form hand gestures in the air above the lamp base, while each photodiode senses, from its vantage point, whether the hand is blocking the light ray emitted by each individual LED on the LED panel. By aggregating the binary blockage information/maps observed from multiple viewpoints (i.e., photodiodes), the system seeks the best-fit 3D hand skeleton in real time using a robust and lightweight reconstruction algorithm.

To realize our approach, we overcome two technical challenges: 1) each photodiode senses only the combined light intensity from all LEDs and ambient light. For a photodiode to recover the blockage information related to individual LEDs, we embed a unique frequency and temporal pattern to the light ray emitted from each LED. Such patterns are imperceptible to human eyes and yet detectable by photodiodes, so that a photodiode can separate light rays and acquire the binary blockage information related to each LED; 2) the search space of 3D hand poses is large, given the high degrees of freedom of the hand. To seek the best-fit hand pose efficiently and robustly, we apply a quasi-random search method [21, 22] to sample the search space. Furthermore, we maintain a window of top candidate poses and infer the final pose as a weighted average of these candidates to achieve robust inference.

We demonstrate the feasibility of our approach by building a proof-of-concept prototype of Aili (Figure 1). The Aili lamp is fabricated following the size of a commercial table lamp [3]. The LED panel comprises 24 × 12 white LEDs and the lamp base is embedded with 16 low-cost photodiodes as a 4 × 4 grid. As a result, each photodiode captures a 2D blockage map with 288 *binary* pixels (each pixel corresponding to the binary blockage information with respect to one LED). The set of 16 blockage maps are used to identify a 3D hand skeleton pose in real time. With Aili, the user can freely gesture under the lamp to navigate and edit virtual 3D objects. Our system evaluation shows that the reconstruction algorithm infers a hand pose within 7.2 ms on average, and achieves an average angular deviation of 10.2° and translation deviation of 2.5 mm in comparison to Leap Motion, a popular commercial hand-tracking system.

Our approach exemplifies the vision that ubiquitous light can be reused as a passive sensing medium to reconstruct gestural input in the 3D space. By augmenting a table lamp with sensing capability, our system reuses existing lighting infrastructure as part of the sensing system at homes and offices, and thus can be seamlessly integrated into the environment, weaving sensing into the fabric of everyday life [63]. Additionally, by relying on such a small number of low-level visual cues (binary blockage pixels), our approach not only entails lightweight computation, but also is inherently privacy-preserving in comparison to camera-based approaches.

Our contributions of this work are: (1) the design and implementation of an Aili prototype that augments a table lamp with the ability to reconstruct hand poses; (2) a real-time reconstruction algorithm that reliably and efficiently reconstructs a hand skeleton using binary blockage maps; (3) a system evaluation of Aili's reconstruction performance across users; (4) user studies to examine the privacy issues of Leap Motion and solicit feedbacks on Aili's privacy protection; and (5) demonstrations of usage scenarios of Aili.

2 RELATED WORK

We categorize existing work on hand gesture sensing based on the sensing medium.

Cameras. Many works use cameras to sense hand poses. We categorize them based on their methodology. The first category of works relies on pre-computed databases and machine learning techniques to find the best-fit hand pose. As examples, 6D Hands [59] uses two web cameras to capture hand images, queries a database of pre-computed 3D hand models to find the pose that best matches hand silhouettes in hand images. It recognizes hand poses at 20 Hz. Hand silhouettes are similar to the blockage maps used in Aili, yet Aili differs in that it does not require any pre-trained databases. With a lightweight pose reconstruction algorithm, Aili's mean reconstruction latency is only 7.2 ms. Similarly in [9], captured hand images are compared to synthetic hand images in a database. In [47], Sridhar, et al. use RGB cameras to capture hand images from different angles and combine databases and machine learning techniques. Depth cameras have also been often used. With hand's depth images, Sharp, et al. build a classifier to recognize hand poses [45], while Keskin, et al. apply multi-layered randomized decision forests [25]. In [52, 53], Tang, et al. further explore variants of regression forest. RetroDepth [27] senses 3D silhouettes of hands using a retro-reflector to separate hands from the background.

A common issue of these systems is the need of a large training dataset and the associated computation overhead. Aili differs in that it recovers the coordinates of hand joints without requiring any database of pre-computed 3D hand models. Instead of directly handling a large number of gray/RGB pixels, Aili captures only hundreds of binary pixels to reconstruct hand poses and entails a lightweight computation.

The second category of works directly computes 3D coordinates of hand joints [10, 13, 15, 26, 37, 38, 41, 48, 51, 54, 56, 61, 64, 70]. These methods commonly represent hand as a 3D hand model and identify the hand pose that best matches hand images by optimizing an objective function. In particular, La Gorce et al. propose an objective function that explicitly uses temporal texture continuity and shading information of the hand [15]. In [10], Ballan, et al. consider hand edges, optical flow, and collisions in the objective function to reconstruct two-hand interactions. Digits [26] uses a wrist-worn optical depth camera to detect how much fingers are bent and applies a kinematic hand model to aid pose prediction. Gradient-based optimization also has been explored for faster convergence. Tagliasacchi, et al. apply a single gradient-based optimization and achieve real-time tracking at 120 Hz [51]. Taylor, et al. construct a smooth-surface model and formulate the problem as gradient-based non-linear optimization [54]. Qian, et al. simplify the hand model using spheres to construct a cost function that combines gradient-based and stochastic optimization [41]. In addition, Oikonomidis, et al. [37] minimize the discrepancy between the hand model projection and the hand image using a variant of particle swarm optimization (PSO). They later introduce an evolutionary quasi-random sampling strategy [38] that speeds up the tracking by 4 times. We are inspired by this work and also apply quasi-random sampling. Our work differs in that with only binary pixels as input, we minimize a different objective function. We also remove the evolutionary part and directly apply quasi-random sampling search, which runs sufficiently well in our system.

Unlike the above works, Aili enables 3D hand reconstruction without cameras, relying on only binary blockage information. Similar to our work, ZeroTouch [34] considers the use of infrared LEDs and sensors for hand poses sensing. However, ZeroTouch only tracks fingers in a 2D plane, while Aili reconstructs 3D hand poses.

Radio Frequency or Acoustic Signals Prior works have also studied the use of radio frequency (RF) or acoustic signals to sense hand gestures. The focus has been on differentiating a small set of pre-defined hand

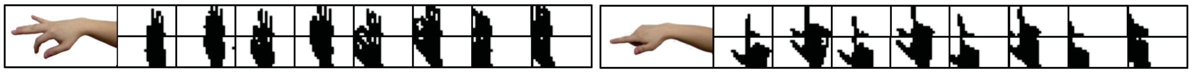


Fig. 2. Binary blockage maps from our Aili prototype for two hand poses. Each pose leads to 16 blockage maps, where each map consists of 288 binary pixels, indicating the blockage information observed by a photodiode in the lamp base.

gestures or tracking a single finger, using Wi-Fi [24, 50**?**], GSM [69], and acoustic signals [18, 43]. In particular, [24] analyzes reflected RF signals to classify eight gestures; [50] tracks the Wi-Fi signal strength and its angle of arrival to track a single finger; [18] leverages the Doppler effect of acoustic signals to identify gestures. Google's recent project Soli leverages 60 GHz signals [5, 60, 67] to recognize subtle finger movements. Aili differs in that it is free from electromagnetic interference and ambient sound interference. Furthermore, it reconstructs arbitrary hand poses and enables fine-grained sensing.

On-Body Sensors Another related line of works relies on sensors worn on user's wrist or fingers to differentiate hand gestures. For wrist-worn sensors, to detect user's forearm shape, prior studies explored the use of capacitive sensors [42], infrared photo reflectors [39, 49], force resistors [16], electrical impedance tomography (EIT) sensors [68], the accelerometer and gyroscope sensors on a smartwatch [65], pressure sensors [33]. Finger-worn sensors include RFID tags [57], a fish-eye imaging device [14], and a ring embedded with an accelerometer and microphone [17]. These systems are limited in the sensing resolution and detect a small set of hand poses (e.g., pinch). In contrast, Aili is device-free and recovers any hand poses.

3 AILI: SYSTEM OVERVIEW

At the high level, Aili reconstructs a 3D hand skeleton based on how the hand blocks light rays emitted by LED chips in the lamp. It captures the blockage information using an array of photodiodes (each 10 mm × 7 mm in size) embedded in the lamp base. Each LED is a point light source emitting light in a cone shape. When the user performs hand gestures under the light, the hand blocks certain LEDs at each given time from each photodiode's point of view. Combining blockage maps collected by different photodiodes, Aili identifies the 3D hand pose that best matches observed blockage maps.

Realizing Aili faces a set of unique challenges. First, detecting the light blockage information is non-trivial using low-cost, off-the-shelf photodiodes. The photodiodes are exposed to multiple light sources including the LEDs in the lamp and the ambient light. Each photodiode perceives only a combined light intensity within its viewing angle. Thus, it is unable to detect which LEDs are blocked by the hand.

Second, our hands are extremely dexterous and flexible. With 23 degrees of freedom, hands can freely move and rotate, generating more complex and subtle hand poses than whole-body postures. Furthermore, fingers are thin and close to one another. Thus they are vulnerable to the occlusion problem. Because of these hand properties, directly applying prior methods [31, 32] on whole-body reconstruction fails to converge at a single hand pose, entails a long reconstruction delay (supporting only 10 FPS), and leads to a poor accuracy.

Finally, each pixel of a blockage map is binary, unlike the RGB/gray-scale pixels in camera images. The number of pixels is small, limited by the LED density and photodiode's limited viewing angle. Furthermore, each blockage map (see Figure 2 for examples) contains only a partial hand projection because of the limited size of the LED panel. All above factors make the pose reconstruction particularly challenging. Most prior reconstruction algorithms using cameras [10, 13, 15, 26, 41, 48, 56, 61, 64, 70] are not directly applicable.

Aili addresses these challenges via two components: *acquiring hand blockage information*, and *reconstructing hand poses using blockage maps*. We next describe them in detail.

4 ACQUIRING HAND BLOCKAGE INFORMATION

Aili's first component is to identify the LEDs blocked by user's hand from each photodiode i's perspective at any given time t. Recovering blockage maps is challenging because photodiode i perceives only the light intensity combining all light rays within its viewing angle. Thus, its light intensity value alone does not suggest which LEDs are blocked.

Aili applies a prior method [31, 32] to embed a unique pattern into the light rays from each LED. In particular, the unique pattern refers to a unique high flashing frequency (20.8 kHz – 40 kHz) imperceptible to human eyes. To support many LEDs with a limited number of flashing frequencies, we further reuse the flashing frequencies across LEDs over time based on the design in [32]. As the photodiode perceives the incoming light intensity over time, it projects the light intensity values within a time window (20 ms) into the frequency domain using FFT. The resulting frequency power at each flashing frequency k is directly proportional to the intensity of the light ray emitted from the LED flashing at frequency k. Thus a significant frequency power reduction indicates the blockage of the corresponding LED. By monitoring the frequency power changes, the photodiode can identify the blockage of each LED separately.

Specifically, given LED j's current frequency power $P_{ij}(t)$ observed by photodiode i, we calculate LED j's frequency power change as $\Delta P_{ij}(t) = |\frac{P_{ij}^{\text{nonBlock}} - P_{ij}(t)}{P_{ij}^{\text{nonBlock}}}|$, where P_{ij}^{nonBlock} is the average frequency power of LED j when no hand is below the lamp[1]. If $\Delta P_{ij}(t)$ is above a threshold δ_{ij}, LED j is considered to be blocked from photodiode i at time t. Similar to the prior design [32], Aili adapts δ_{ij} based on the light intensity I_{ij} normalized to the maximal light intensity I_{max} among all light rays. Thus, we set δ_{ij} as:

$$\delta_{ij} = P_{min} + (P_{max} - P_{min}) \cdot \frac{I_{ij}}{I_{max}}, \tag{1}$$

where P_{min} and P_{max} are the minimal and maximal $\Delta P_{ij}(t)$ (0.7 and 0.4 in Aili). Aggregating the blockage detection results for N LEDs leads to the blockage map $S_i(t)$ at photodiode i as: $S_i(t) = \{s_{ij}(t)|0 < j \leq N]\}$, where $s_{ij}(t)$ indicates whether the light ray fro LED j to photodiode i is blocked. We have $s_{ij}(t) = 1$ when $\Delta P_{ij}(t) > \delta_{ij}$ and $s_{ij}(t) = 0$, otherwise. Figure 2 shows example blockage maps recovered at 16 photodiodes for two hand poses. In the next section, we describe how to leverage these blockage maps to reconstruct 3D hand poses.

5 RECONSTRUCTING HAND POSES

As the main technical contribution of our work, the second component of Aili reconstructs fine-grained 3D hand poses using only 2D hand blockage maps with *binary* pixels. We break down the reconstruction into two steps:

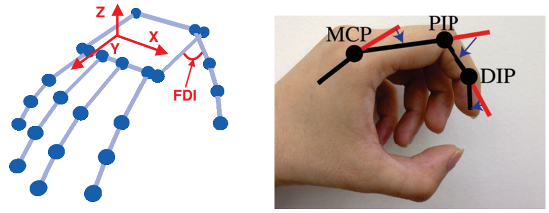
(1) We first locate the hand in the 2D plane based on coarse hand features extracted from the current set of blockage maps. We consider the wrist center and the first dorsal interossei (FDI) next to the thumb as reference points indicating hand's coordinates in X and Y axis (Figure 3(a)).
(2) We then search for the hand pose (described by a 3D hand model, Figure 3(a)) and hand height (Z-axis coordinate) that best match the blockage maps. We formalize it as an optimization problem, seeking to minimize the mismatch between the candidate hand pose and the blockage/non-blockage information revealed by blockage maps.

Since the 2D tracking (Step 1) does not leverage the prior 3D reconstruction result, it is not affected by reconstruction errors and avoids errors to be accumulated. Furthermore, by only relying on the current blockage maps to conduct the 2D tracking, Aili also prevents prior tracking errors from propagating to the current reconstruction result. Next, we first present a hand kinematic model that characterizes the dependency of finger joints and biomechanical constraints of human hands. The model reduces the number of finger joints to track. We then describe the two steps in detail.

[1] We measure P_{ij}^{nonBlock} in the beginning of each experiment.

5.1 Hand Kinematic Model

As illustrated in Figure 3(a), we represent a hand pose using a set B of 19 segments. They include 1) a set B_F of 15 finger segments, where each finger contains three segments connected by finger joints, and 2) a set B_P of 4 palm segments that describe the palm contour (a rectangle). Our current design assumes that user's hand is measured beforehand and hand parameters (e.g., finger length, palm width) are known. Aili's reconstruction algorithm then is to search for the set of finger and palm segments that best match the blockage maps observed by all photodiodes. Given the large space of possible finger and palm configurations, we apply a hand kinematic model to reduce the computational complexity in searching. The kinematic model defines the natural interdependency of the finger joints, allowing us to use one of the joint's flex angle to extrapolate how much the other joints are naturally bent on the same finger [11]. Figure 3(b) marks the three joints of the index finger (e.g. the metacarpophalangeal joint (MCP), proximal interphalangeal joint (PIP), and distal interphalangeal joint (DIP)). Among these joints, if we know the flex angle of the MCP, we can infer the flex angles of the PIP and DIP by using the following equations [11, 23, 26]: $\theta_{PIP} = \frac{\theta_{MCP}}{0.54}$, $\theta_{DIP} = \frac{\theta_{MCP} \times 0.84}{0.54}$, where θ_{MCP}, θ_{PIP}, and θ_{DIP} are the flex angles of the MCP, PIP, and DIP, respectively. By leveraging this simple kinematic model, we are able to reduce the degrees of freedom of the hand from 23 to 15 without affecting the reconstruction accuracy. Furthermore, this kinematic model also helps produce hand poses that are natural and subject to human hand's biomechanical constraints. Figure 3(a) is a hand pose reconstructed by Aili when the user is naturally opening the fist.



(a) Hand skeleton in Aili (b) Kinematic model

Fig. 3. Hand skeleton model in Aili. (a) We represent a hand pose using 15 finger segments (3 segments per finger) and 4 palm segments outlining the palm contour. (b) Finger joints on the same finger are interdependent during movement.

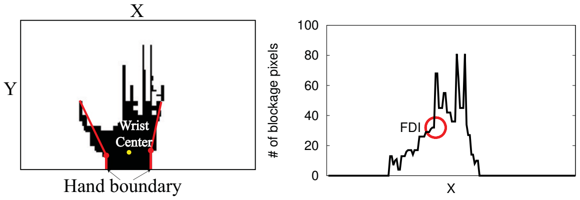
5.2 Tracking Hand's 2D Location

Tracking the hand position in a 2D plane can be done by tracking a number of distinguishable hand features that are insusceptible to the change of hand poses. Our current implementation uses two hand features: the center of the wrist and the first dorsal interossei (FDI), marked in Figure 3(a).

Aggregating Blockage Maps Feature extraction is particularly challenging because of pixel's binary nature and the low resolution of blockage maps (24 × 12 pixels). Additionally, blockage maps contain only a partial hand given the relatively small field-of-view of photodiodes. To solve the problem, we aggregated all 16 blockage maps at a given time to obtain a complete image of the hand. Specifically, since black dots in blockage maps represent blocked light rays, we leverage a horizontal plane at the hand height of the last reconstruction result to locate the intersections of blocked light rays on the plane. These intersection points represent the projection of the hand shape. By aggregating all intersection points into a blockage map, we can acquire more information on the hand shape and extract coarse hand features. Note that the initial height of the hand is unknown when the hand is first registered to the system (e.g. at the beginning of a gesture). We require users to start with an open-fist pose as a gesture delimiter. The system can then discover the hand position by permuting all possible positions in a 3D space. This process takes roughly 30 ms on a Dell T5500 server. Figure 4(a) shows an example of the aggregated blockage map.

Extracting Hand Features We detect the wrist center by first scanning the hand contour from the bottom to halfway towards the upper bound of the contour (the scanned contour is marked as red lines in Figure 4(a)). We identify the wrist by seeking a pair of inflection points with the greatest curvatures, the center of which is considered to be the wrist center.

Next, the FDI can be identified by first counting the number of the blockage pixels in each column of the aggregated blockage map. In the resulting histogram (see Figure 4(b)), we then identify the FDI by looking for the first point with a first-order derivative greater than a threshold value (e.g. 20). It is worth mentioning that the FDI is more accurate in tracking the hand position in the Y axis. However, this feature may disappear when the thumb is bent towards the palm. Therefore, we use the wrist center as the primary feature in tracking the hand position while the FDI is only used a secondary feature to assist the tracking in the Y axis (marked in Figure 3(a)). With this simple method, the 2D tracking error is within a few millimeters. Such high 2D tracking accuracy is essential to the later reconstruction.



(a) Wrist center (b) First dorsal interossei (FDI)

Fig. 4. Extracting two hand features (wrist center and the FDI) from the aggregated blockage map.

5.3 Determining Hand Pose and Height

Given the hand's coordinates in the 2D plane, we now seek the best-fit hand pose and hand height. For a candidate hand height, we solve the hand pose reconstruction as an optimization problem. We define the objective function $E(B)$ to evaluate the mismatch between a candidate hand pose and the set of blockage maps at time t. In particular, a candidate pose is represented by the 3D hand skeleton model B (Figure 3(a)) and we calculate $E(B)$ as:

$$E(B) = \sqrt{a \cdot E^2_{block}(B) + b \cdot E^2_{unBlock}(B)}, \tag{2}$$

where $E_{block}(B)$ is a penalty count for blocked light rays. It increases when a candidate hand pose fails to block a light ray that is supposed to be blocked according to the blockage maps. $E_{unBlock}(B)$ is the penalty count for unblocked light rays. It increases when a candidate hand pose blocks a light ray that is not supposed to be blocked according to the blockage maps. The coefficients a and b represents the ratio between the blocked and unblocked light rays in the current blockage maps. We aim to minimize both $E_{block}(B)$ and $E_{unBlock}(B)$ so that the user's hand poses can be best recovered. Ideally, both $E_{block}(B)$ and $E_{unBlock}(B)$ are close to 0 when the best match is found. Combining both the blockage and non-blockage constraints enhances Aili's ability to filter out ambiguous candidate hand poses caused by the finger occlusion. It also helps the search algorithm converge at the best-fit hand pose more quickly.

Computing $E_{block}(B)$ and $E_{unBlock}(B)$ takes three steps:

(1) We first gather blockage maps from all photodiodes at time t and identify all blocked light rays, denoted by the set L_1. The remaining light rays are unblocked, denoted by L_2.

(2) Next, we examine how light rays intersect a candidate hand pose. We consider that a light ray is blocked if it intersects any finger B_F or the palm B_P in the 3D hand model. To determine the intersection with a finger segment $b_m \in B_F$, we compute the perpendicular distance between the light ray and b_m. We examine whether the distance is shorter than the radius of the finger segment cylinder. To determine the intersection with a palm B_P, we examine whether the light ray passes the rectangle area defined by the four palm segments.

(3) Finally for each blocked light ray $l \in L_1$ that does not intersect any finger segments or the palm rectangle, we determine its penalty as its distance to the closest finger or palm segment. We set the penalty as the minimal distance to a segment because the blocked light ray does not need to block all finger segments and the palm. l's penalty is zero if it does intersect any finger segment or the palm. Therefore, we can write $E_{block}(B)$ as:

$$E_{block}(B) = \sum_{l \in L_1} \min_{\substack{b_m \in B_F \subset B \\ B_P \subset B}} (d(l, b_m), d(l, B_P)),$$

where

$$d(l, b_m) = \begin{cases} dist(l, b_m) - r_m & \text{if } dist(l, b_m) > r_m \\ 0 & \text{otherwise} \end{cases}$$

$$d(l, B_P) = \begin{cases} 0 & \text{if } l \text{ intersects palm } B_P \\ \min_{p_n \in B_P}(dist(l, p_n)) & \text{otherwise} \end{cases}$$

in which, b_m and p_n is a finger and palm segment of a candidate hand pose B (Figure 3(b)), respectively, $dist(l, x)$ is the distance between light ray l and a (finger/palm) segment x, and r_m is the radius of the finger segment cylinder b_m.

Similarly, for each unblocked light ray $l' \in L_2$ that intersects either a finger segment or the palm rectangle, its penalty is its maximal distance to escape all finger segment cylinders or the palm that it intersects. We take the maximal distance as the penalty here because the unlocked light ray is not supposed to intersect any finger segments or the palm. It also ensures that the penalty of l' is zero only if l' intersects neither finger segments nor the palm. Thus, $E_{unBlock}(B)$ is written as:

$$E_{unBlock}(B) = \sum_{l' \in L_2} \max_{\substack{b_m \in B_F \subset B \\ B_P \subset B}} (d'(l', b_m), d'(l', B_P)),$$

where

$$d'(l', b_m) = \begin{cases} r_m - dist(l', b_m) & \text{if } dist(l', b_m) < r_m \\ 0 & \text{otherwise} \end{cases}$$

$$d'(l', B_P) = \begin{cases} \min_{p_n \in B_P}(dist(l', p_n)) & \text{if } l' \text{ intersects palm } B_P \\ 0 & \text{otherwise} \end{cases}$$

Therefore, our goal is to find the best-fit $B^\star$ that minimizes the objective function (Eq. (2)) for the current candidate hand height: $B^\star = \text{argmin}_{B \in \mathbb{B}} E(B)$, where $\mathbb{B}$ denotes the search space of all possible hand poses. The challenge lies in dealing with the large search space and the discontinuity of our objective function $E(B)$, which renders gradient-based optimization [41, 51, 54] not applicable. Thus, we have focused on exploring sampling-based methods, including a sequential search method with a fixed step size, heuristic sampling methods such as particle swarm optimization (PSO) [37], as well as quasi-random sampling methods [38, 40]. We decide to choose quasi-random sampling as our final method because it entails the lowest computational overhead and does not require a large number of samples to achieve high accuracy. In comparison, PSO's efficacy heavily depends on the number of particles and evolutionary generations. Later in Section 7.2, we will also compare the performance of these algorithms. We next describe our search algorithm in detail.

Quasi-Random Search for Hand Poses Quasi-random sampling constructs sequences of D-dimensional points that are almost uniformly distributed in the hypercube $[0, 1]^D$ [46]. These sequences are also called low-discrepancy sequences, because for any subset of the hypercube, the number of sampling points it contains is nearly proportional to its volume [36]. Because of this property, quasi-random sampling covers a high-dimensional space more uniformly and quickly than pseudo-random sampling. It has been used in Monte Carlo integration [40] and to speed up hand tracking [38]. There are several methods to construct quasi-random sequences, such as Sobol, Halton, Faure and Niederreiter family of sequences [35]. We adopt Sobol's sequence since it performs well in moderate dimension space [12]. We construct the Sobol sequence in gray code order, following the method in [21, 22]. We apply different scales to different parameter dimensions of a Sobol point, because Sobol sequence is in $[0, 1]^D$ space and different parameter tends to have different change rate during an inference (i.e., a frame).

To infer the hand pose $B_t^\star$ for the t-th frame, we generate M candidate poses around the pose $B_{t-1}^\star$ of the previous frame based on the Sobol sequence. Among the M candidates, instead of simply picking the pose with the minimal $E(B)$ value, we compute a weighted average of the top $M_T(< M)$ candidates (ranked in ascending order of their $E(B)$ values). The averaging makes the pose inference more robust against noises and multiple local minimums. Algorithm 1 lists the outline of our algorithm, where we maintain M_T candidates along with their $E(B)$ values in a priority queue Q. We implement Q as max heap to facilitate the tracking of top-M_T candidates during the search. Finally, we infer $B_t^\star$ as the weighted mean of the M_T candidates in Q:

ALGORITHM 1: Quasi-Random Search for Hand Poses

```
input  : Inferred hand pose $B_{t-1}^\star$ of $t-1$-th frame
output : Inferred hand pose $B_t^\star$ of $t$-th frame
begin
    $Q \longleftarrow \emptyset$
    insert $M_T$ arbitrary hand poses with key of INFINITY into $Q$
    for $B_c \in SobolPoints(B_{t-1}^\star, M)$ do
        $q \longleftarrow Top(Q)$
        if $E(B_c) \leq q.key$ then
            $Remove(Q, q)$
            $Insert(Q, \{E(B_c), B_c\})$
        end
    end
    $B_t^\star \longleftarrow WeightedMean(Q)$ (Eq.( 5))
end
```

$$B_t^\star = \frac{1}{\sum_{B_c \in Q} \frac{1}{\log E(B_c)}} \sum_{B_c \in Q} \frac{B_c}{\log E(B_c)}. \tag{5}$$

We further accelerate the quasi-random search by reducing the search space. We infer first the hand position[2] and then finger joints. We partition five fingers into two groups (thumb and index fingers in a group, while others in another group) and optimize only one group in an inference. Specifically, after applying the hand kinematic model, we optimize thumb and index fingers (5 degrees of freedom) in odd frames and other three fingers (6 degrees of freedom) in even frames.

The efficacy of our algorithm relies on proper configuration of several key parameters: the scaling vector of hand pose parameters, the averaging window size M_T, the total number M_p of sampled Sobol points for a hand position, and the number M_f of sampled Sobol points for finger joints. After testing different scaling vectors, we divide the scaling vector of hand pose parameters into four types: the positional scale (1 cm), the thumb joint angle scale (10°), the MCP pitch angle scale (25°), and the MCP yaw angle scale (1°) of the remaining fingers. We test different M_T and set it as 5. To configure M_p, M_f, we compare the distribution of angular errors under different M_p, M_f using simulations and plot results in Figure 5. We observe that once $M_f \geq 32$, the improvement of accuracy becomes marginal in both the mean and the tail. Since larger M_p and M_f entail longer latencies, we choose $M_p = 8, M_f = 32$ to achieve the best tradeoff.

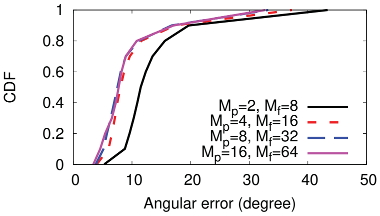
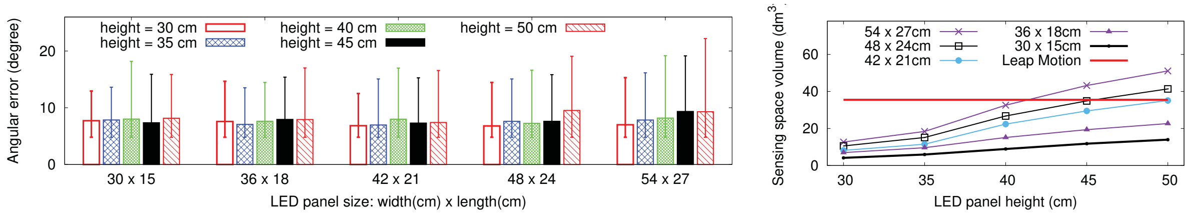


Fig. 5. Cumulative distribution functions (CDF) of angular errors under different M_p and M_f.

6 AILI PROTOTYPE

We build an Aili prototype using off-the-shelf LEDs, low-cost photodiodes (<$12), and micro-controllers (e.g., Arduino Due). While aiming for the optimal reconstruction performance, we also bear in mind design considerations for Aili to look and function like a regular lamp. Next, we elaborate on the physical design and hardware implementation.

[2] The hand position has only 2 degrees of freedom since the position on the Y axis has already been decided in Section 5.2.



(a) Reconstruction accuracy

(b) Sensing area size

Fig. 6. Examining the impact of LED panel size on reconstruction accuracy and sensing area size, using the Aili emulator.

6.1 Physical Design of Aili

The following choices are key to Aili's physical design: the LED panel size and height, the density of LED chips, and the panel shape. We aim to seek the configuration optimizing sensing performance while ensuring Aili's original function as a lamp.

Aili Emulator in Unity To avoid experimenting numerous possibilities of hardware configurations, we build an Aili emulator using Unity5, a popular game engine that can precisely simulate light ray propagation using ray casting. We set up a virtual scene with a virtual LED panel, containing a configurable number of point light sources (i.e., LEDs), a 3D hand model from [4], and a horizontal surface representing a tabletop. On the table, we place 16 virtual photodiodes as a 4 × 4 grid in a 21 cm × 16.5 cm area.

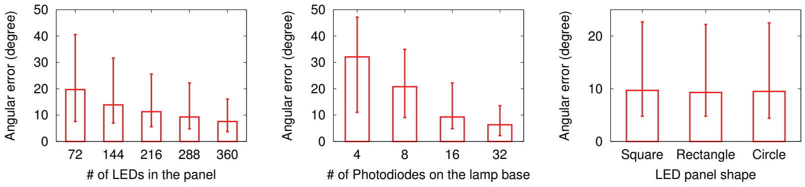
We write a Unity program that controls the virtual hand to perform a set of gestures in Figure 8 and their combinations (e.g., a combined gesture of Figure 8(a) and Figure 8(i)) at three height levels (15 cm, 20 cm, and 25 cm) above the virtual tabletop. For a given configuration, our program uses a ray-casting algorithm to cast a light ray from an LED to a photodiode. It then detects light rays that are blocked by the virtual hand and estimates the blockage map observed by each photodiode. We then run our reconstruction algorithm with these estimated blockage maps to generate a hand pose, based on which we compute the angular error of each finger segment.

We validate the accuracy of our emulator by comparing it to the Aili prototype (§ 6.2). We 3D print the virtual hand, place it at 9 locations in Aili's sensing area, and compare estimated blockage maps and reconstructed hand poses to that from the emulator. From our results, estimated blockage maps closely match actual maps (differing in 4.5% of pixels), and the mean angular deviation of the reconstructed skeletons is 0.2°. The results justify our use of the emulator to examine the impact of Aili's design parameters summarized as below.

LED Panel Size and Height We start with testing the LED panel size and height. Simulating Aili with 288 LEDs, we vary the panel size and height within the range of normal table lamps. Figure 6(a) compares the angular error of reconstruction results when the hand moves within Aili's sensing area. We also include 90% confidence intervals as error bars. We define the sensing area as the 3D space where the mean angular error is no larger than 12° (the threshold we identified in a pilot study). We observe that Aili's accuracy is relatively stable across panel sizes and heights. The reason is that with a fixed number of LEDs and photodiodes, panel size and height do not affect the number of light paths used by Aili to recover blockage maps, as long as the hand moves within the sensing area. As a result, the reconstruction accuracy is similar across these height and size configurations.

We further examine the impact of the LED panel size and height on the sensing area size, which largely affects the system usability. Figure 6(b) compares the sensing area size under various LED panel configurations. We also plot Leap Motion's sensing area size for reference. We measure Leap Motion's sensing area by eyeballing whether there is any visual difference between the actual and reconstructed hand poses using the default Leap visualizer application [6]. We make two observations. First, as expected, a higher or larger LED panel results into a larger sensing area. It is because a larger and higher LED panel spreads light rays in a larger space, enlarging the area where hand blockage can be captured. Second, Aili outperforms Leap Motion in sensing area size when the panel is sufficiently large and high (e.g., a 48 × 24 cm panel at the height of 45 cm). In particular, a 54 × 27 cm panel at 50-cm height achieves a sensing area 50% larger than that of Leap Motion. We choose this configuration to build the prototype for the maximal flexibility and ease to test a variety of hand poses.

LED/Photodiode Density and Panel Shape We move on to testing LED density and panel shape. Figure 7(a) plots the accuracy when we fix the panel size (54 cm × 27 cm) and vary the number of LEDs. As expected, denser LEDs reduce the angular error because blockage maps contain more pixels, which lead to more detailed hand contour. The caveat of a high LED density is that to keep the overall lamp illumination within a usable range (below 2000 lx [7, 8]), each LED's illumination needs to be sufficiently low, which makes it hard for the photodiode to detect light change from individual LEDs. To strike a better balance, we choose 288 LEDs. We then vary the photodiode density. Figure 7(b) plots the accuracy when we fix the LED density (288 LEDs on 54 cm × 27 cm panel) and vary the number of photodiodes. We observe that as the number photodiodes increases, they provide more diverse blockage maps to represent a 3D hand pose and improve the reconstruction accuracy. The downside is the increase in reconstruction latency to deal with more blockage maps. We choose 16 photodiodes that achieve a good tradeoff. Finally, we vary panel shape while fixing the panel size, density of LEDs and photodiodes. We observe negligible differences across shapes (Figure 7(c)). We choose to build the prototype in a rectangular shape for the ease of fabrication.



(a) LED density (b) Photodiode density (c) Panel shape

Fig. 7. The impact of LED/Photodiode density and panel shape on reconstruction accuracy, evaluated by the Aili emulator.

6.2 Aili Hardware Component

LED Panel We build a customized LED panel 54 cm × 27 cm in size and mount it inside a customized lampshade at 50-cm height (Figure 1). The panel consists of 12 Printed Circuit Boards (PCBs) pieced together using 3D-printed plastic connectors. Each PCB board contains 6 × 4 LED chips (Cree U2) with a 2.25-cm interval, MOSFET, resistors, and capacitors. The PCB is made of aluminum to dissipate the heat generated by LED chips. When all the LEDs are on, the temperature is 56°C at the panel surface and 37°C at 1 cm away from the panel surface. Thus, the hand does not experience heat from the panel once it is a few centimeters away. Each PCB board is connected to an FPGA (Digilent Basys 3) and driven by an individual power supply (4.5 V). The 12 FPGA boards are arranged in two layers on the panel back. We implement the prior design [32] on FPGA boards to modulate LED's flashing rates, which range from 20.8 KHz to 40 KHz to avoid any flickering effect [28, 30]. The panel's illumination is measured as 1900 lx on the table right below the panel center.

Photodiodes We arrange 16 photodiodes (OPT101) in a 4 × 4 grid in a 21 cm × 16.5 m area in the lamp base (Figure 1). We select OPT101 for three reasons. First, it is highly responsive to small light changes (0.45 A/V for 650-nm wavelength). Second, its bandwidth (e.g. 56 KHz) is sufficient to support the highest LED flashing rate (40 KHz). Third, it has a relatively wide viewing angle (140° on x-axis and 100° on y-axis) ensuring that all LEDs are visible to the sensor. The resulting sensing space is 51 dm^3 in volume above the table.

We connect each photodiode to a 50-KΩ resistor and 56-pF capacitor in series on a customized PCB board to avoid sensor saturation. Each photodiode is driven by an Arduino DUE board, which measures the resistor's voltage to infer the light intensity at the photodiode. 16 Arduino boards are connected to a server (a Macbook Pro 13-inch laptop) through serial ports, where the server then generates blockage maps and runs our reconstruction

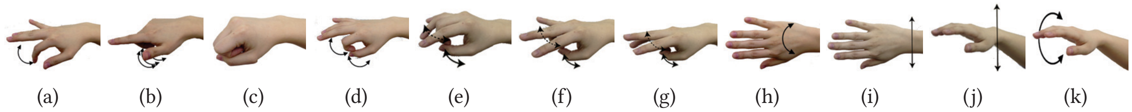


Fig. 8. The eleven test gestures in the experiment: (a) bending the index finger, (b) bending the middle, ring, and little fingers simultaneously, (c) closing the fist, (d) – (g) pinching the thumb with the index, the middle, the ring, and the little finger, respectively, (h) waving palm around the wrist, (i) moving the hand horizontally, (j) moving the hand vertically, and (k) rotating the wrist in a circle.

algorithm to infer hand poses. To overcome Arduino's limited processing power, we implement a processing pipeline similar to [32], which allows blockage maps to be generated in 20 ms.

7 SYSTEM EVALUATION

We evaluate Aili by inviting another group of participants to test our Aili implementation. We aim to understand both the system performance and user's experiences of using Aili. To examine Aili's system performance, we compare Aili to Leap Motion because of its accuracy and popularity. We understand that Leap Motion's sensing performance is not perfect (as shown in our later study). We thus treat it as a benchmark rather than ground truth. We examine Aili's reconstruction accuracy, latency, and the impact of lighting condition.

7.1 Experimental Setup

Participants We recruited 10 participants (3 females and 7 males) between ages of 20 and 30. They are right-handed and daily computer users. Their hand size varies from 7 cm to 9 cm in width, 7 cm to 8.5 cm in length.

Apparatus Apparatuses include the Aili prototype and a Leap Motion sensor. We place the Aili lamp on a regular computer desk. Given Leap Motion's limited working range, we place it in the center of the lamp base for Leap Motion to perform the best, where the participant's hand hovers above the Leap Motion. Leap Motion sensor emits strong infrared signals that interfere with the photodiodes in Aili. Thus, we cover the photodiodes with infrared filter lens, which help largely remove the infrared noise.

Task and Procedure Prior to the study, we inform each participant of the study purpose. The participant has the opportunity to ask questions about Aili. We then measure the participant's right hand and feed their hand parameters (e.g., palm width, finger length) into the system. During the study, we instruct each participant to use the right hand to perform 11 hand gestures, which include bending the index finger, bending the middle, ring, and little finger together, making a fist, pinching the thumb with the index finger, pinching the thumb with the middle finger, pinching the thumb with the ring finger, and pinching the thumb with the little finger, followed by waving the hand in four different ways, including ulnar/radial deviations, left and right horizontally, up and down virtually, and drawing a circle in the virtual plane. Figure 8 illustrates all test gestures. Participants perform these gestures with hands at their comfortable heights, ranging from 8 cm to 34 cm above the table.

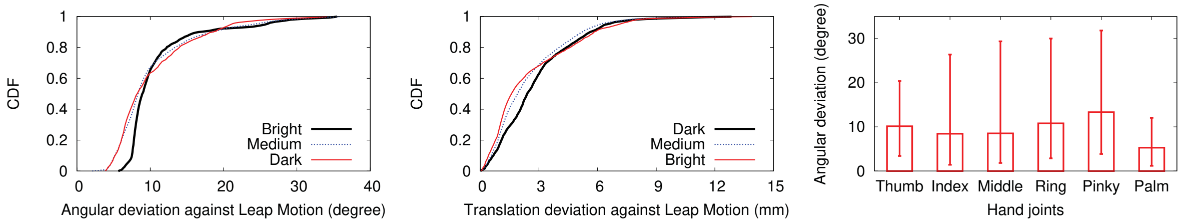
Each participant performs these gestures continuously without any break. This is to emulate the real-world usage scenario, where the user may want to perform a series of continuous gestures. Participants are not asked to rigidly hold the hand in parallel to the table, their fingers can tremble, and they can move their hand anywhere within the sensing area. During the study, the participant can either sit on a chair and place the elbow on the desk to gesture or stand up with arms dangling under the lamp. The participants perform all gestures following the order in Figure 8. we repeat this process three times and record hand motion data for analysis. For each repetition, we examine one of these ambient light conditions: 1) *Dark condition* (0 lx – 20 lx) emulating the night or a dark room where we turn off all lights and close the window blinds; 2) *Medium-light condition* (80 lx – 120 lx) emulating a cloudy day, where we open the window blinds to allow natural sunlight in; 3) *Bright condition* (200 lx – 300 lx) emulating a sunny day, where we turn on the indoor lighting and open the window blinds. Finally, we simulate the walk-up-and-use condition by asking the participants to perform the gestures without any training.

After the study, participants are invited to test two demo applications: 1) controlling the pose of a 3D hand model (Figure **??**(b)); and 2) navigating Google Earth using double-click to zoom in and a close-hand to rotate the earth (Figure **??**(a)). At the end of the study, participants complete a questionnaire for subjective feedbacks.

7.2 Results

We report Aili's accuracy and latency. Statistical analysis is conducted using Repeated Measures ANOVA.

Reconstruction Accuracy We evaluate the reconstruction accuracy using two metrics: *angular deviation* and *translation deviation.* The angular deviation measures the angular difference between the 14 finger segments (represented as 3D vectors) generated by Aili and that by Leap Motion. The angular deviation of the palm is measured based on the palm vector, which starts from the wrist center to the palm center. Translation deviation measures the difference in the hand model's movement trajectories (represented by the wrist's coordinates in x, y, and z axis), reconstructed by Aili and Leap Motion. Figure 9(a) and (b) plot the cumulative distribution functions (CDF) of the angular and translation deviation under the three ambient light conditions.



(a) Angular deviation

(b) Translation deviation

(c) Accuracy of varying hand parts

Fig. 9. Aili's performance under varying ambient light conditions. We also analyze the accuracy across fingers and the palm.

Overall, the average angular deviation between Aili and Leap Motion is 10.2° with the 95th-percentile at 24°. The average translation deviation is 2.5 mm with the 95th-percentile at 6.2 mm. We observe that gestures that involve small finger movement (e.g., little finger movement in Figure 8(g)) cause high deviation errors. Smaller fingers (e.g., little finger) introduce less blockage information than larger fingers (e.g., index finger). Thus, reconstructing smaller finger movement is more challenging given the limited LED/photodiode density. We also observe that some large angular and translation deviations are due to the imperfections of Leap Motion, where the hand pose reconstructed by Aili is actually closer to the actual pose. Figure 10 lists two examples.

We also analyze Aili's accuracy across fingers and the palm. Figure 9 (c) shows the angular deviations of fingers and the palm. A repeated measures ANOVA reveals a significant effect of finger ($F_{5,45} = 25.9, p < 0.001$). A post-hoc analysis with Bonferroni corrections shows that the palm vector has the lowest angular deviation (5.3°, all $p < 0.05$), because palm is the largest part of the hand and easier to track. The little finger (13.3 °, sd = 2.7) has the highest angular deviations (all $p < 0.05$), mainly because of its smallest size, making it less identifiable in blockage maps. We find no significance between thumb and little fingers ($p = 1$) and between the index (8.4°, sd = 2.4), middle (8.5°, sd = 1.6), and ring finger (10.8°, sd = 1.8) (all $p > 0.2$).

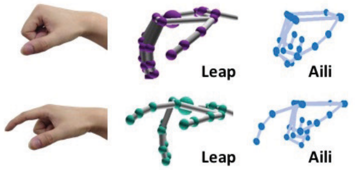


Fig. 10. Examples where Aili outperforms LeapMotion in reconstructing hand poses.

Influence of Ambient Light As we compare the results across different ambient light conditions, the ANOVA shows no significant effect of the lighting conditions on both angular deviation ($F_{2,18} = 2.4, p = 0.12$) and translation deviation ($F_{2,18} = 0.05, p = 0.96$). This result is expected, because ambient light fluctuates randomly, resulting in the frequency power close to the DC component, far outside the frequency range of the LED's flashing rates (20 kHz – 40 kHz). By extracting the frequency powers only at LEDs' flashing frequencies, Aili automatically filters out the ambient light inference. Thus Aili is robust against ambient light variations, supporting its practical use in diverse scenarios.

Reconstruction Latency Next, we examine Aili's reconstruction latency for generating a hand pose. We measure the latency by logging the timestamp of the reconstruction algorithm and plot the CDF of latency in Figure 11. The latency varies from 3.6 ms to 16.7 ms, mainly depending on the number of blocked light rays. For hand poses blocking more light rays (e.g., open hand), the algorithm computes more distances between the hand model and blocked rays to optimize $B^{\star}$, resulting into larger latencies. Overall, the reconstruction latency is 7.15 ms (140 Hz) on average with 95th-percentile at 8.13 ms (120 Hz), thanks to its lightweight search algorithm using a small number of binary pixels. Note that the algorithm is run only on CPU. With GPU acceleration in the future, the reconstruction latency can be further reduced. We also compare Aili's running time to Leap Motion. Aili takes about 40% CPU usage on a laptop, while Leap Motion takes roughly 50% CPU usage even with its specialized hardware augmentation. Overall the result suggests that Aili can be used for real-time interaction.

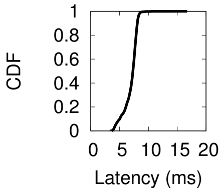
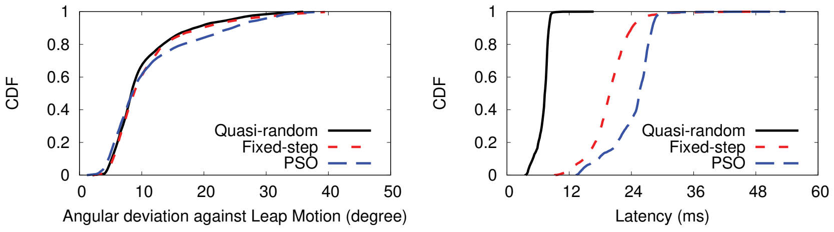


Fig. 11. The latency of reconstructing a hand pose in Aili.

Comparison of Reconstruction Algorithms We also compare the accuracy and latency of our algorithm to other sampling methods, i.e., fixed-step sequential search and particle swarm optimization (PSO). In particular, the fixed-step method sequentially infers parameters of hand's position, the thumb, index, and other fingers, rather than examining all possible combinations. It examines candidate poses only within a small range (i.e., at most ±20° for finger joints, ±1 cm for hand position) of the previous hand pose. To generate a candidate pose, it adjusts a finger joint or hand position by a fixed step at a time (5° for finger joints, 0.25 cm for hand position). To speed up the search, we also divide fingers into two groups (e.g., thumb and index fingers in one group, while others in the other group), similarly to our algorithm. We update the groups at different rates. For the PSO method, we generate 15 particles by simultaneously perturbing finger joints and hand position based on the previous hand pose. The perturbation range is the same as that of the fixed-step method. We then optimize particles for 10 generations.

Figure 12 shows CDFs of angular deviation and latency of these methods. The reconstruction accuracy is similar across methods, where the mean angular deviation is 10.25 °, 10.85° and 11.14° for our algorithm, fixed-step search, and PSO, respectively. However, our algorithm runs much faster. It means latency is 32% and 30% of that of the fixed-step search and PSO method. It also reduces the 95th-percentile latency from 25.5 ms (fixed-step) or 28.5 ms (PSO) to 8.1 ms. The result demonstrates that our quasi-random search algorithm speeds up the reconstruction by 3 times without sacrificing accuracy.



(a) Accuracy (b) Latency

Fig. 12. Comparison our quasi-random search algorithm, fixed-step sequential search, and PSO in accuracy and latency.

Subjective Feedback All the participants felt that Aili's sensing region was large enough for comfortable use (4 with 5 being strongly agreed). They were satisfied with the brightness of the lamp (3 with 5 being too bright). Participants liked the height of the lamp (3.8) and thought that the height of our prototype was about normal for any table lamp. Finally, participants expressed the need for different styles and colors so that the lamp could fit nicely in their home.

8 ELICITED USER FEEDBACK ON PRIVACY PROTECTION

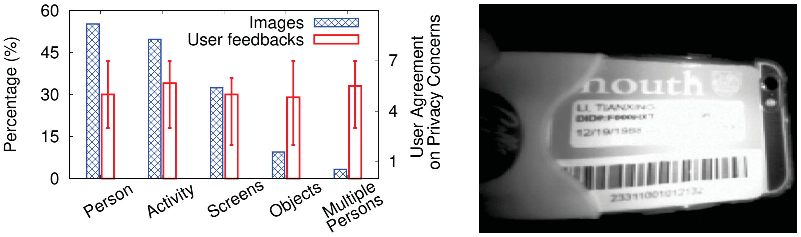
A side benefit of Aili is its inherent privacy-preserving nature, as it captures only the *binary* blockage information of a small number of pixels. In comparison, although camera-based approaches achieve high sensing accuracy, camera images can be leaked to the adversary [44, 62] and impose privacy risks [19]. Even if such privacy risks can be mitigated by various techniques (e.g., disabling cameras when they are not used, processing images locally without storing raw images), malware at firmware or software can still perform targeted attacks by hijacking cameras, as shown in prior studies [55, 66]. In this section, we conduct user studies to examine user feedback on a camera-based hand-motion tracking system (Leap Motion) and Aili on privacy protection.

8.1 Leap Motion Observational Study

Prior studies have revealed the privacy issues introduced by cameras in the context of wearable cameras [19] and cameras on mobile devices (e.g., laptops, smartphones) [55, 58, 66]. In the context of desktop hand-gesture tracking systems, however, it is still unclear whether privacy issues exist, since many of them (e.g. Leap Motion) have cameras facing the ceiling rather than users.

To examine this issue, we conducted a week-long observational study of Leap Motion. We invited six volunteers (22-30 years old, one female). All participants have used Leap Motion before. Half of them use desktops and the other use laptops. Participants were asked to put the Leap Motion on their desks in its best working position (e.g. in front of the keyboard or screen). They can move it if the device affects their work. A participant placed the device on top of his monitor with the device's cameras facing the table. A Python program running on their computers collects data from Leap Motion's built-in cameras. While video recording is possible, we only recorded images (1 per second) to save storage space. Participants were asked to run the program for at least an hour per day. They were not informed of the study purpose during the study.

Two inspectors manually labeled each image to identify events that may raise privacy concerns, including revealing faces, activities, computer screens, personal items, and multiple people [19]. Participants were then invited to complete a questionnaire, asking their agreement on privacy concerns after seeing three randomly selected images in their data set from each category. Participants were informed of the fraction of each category in the data set. Ratings were from 1 to 7 on a continuous numeric scale (1 strongly disagree, 7 strongly agree) with decimal ratings like 3.5.



(a) Study results

(b) Example image

Fig. 13. Study on images captured by Leap Motion. (a) shows percentages of image categories and user feedbacks (error bars show 90% confidence intervals). (b) is an example image, revealing a student ID card stored in the card holder on the back of a smartphone.

We collected approximately 10 hours of data per participant (195,395 images in total), among which 70% contained objects that are previously reported as the source of privacy concerns [19]. In particular, 55% of them contained persons, 49% contained users' activities (e.g. working, drinking yogurt, and laughing), 32% contained computer screens, 9.5% contained objects (e.g. working tools, smartphones, and student IDs), and 3.3% contained multi-person activities (Figure 13(a)). In general, participants expressed privacy concerns about Leap Motion (5.8, SD = 1.2). A repeated measures ANOVA revealed no significant effect of the category in user responses ($F_{4,20} = 0.396, p = 0.81$), indicating that reducing the likelihood of revealing an object in images did not mitigate users' privacy concerns.

An unexpected finding is that we were able to reveal student ID cards, stored in the card holder on the smartphone back (Figure 13(b)). Student ID cards contain personal information and are often linked to financial accounts (e.g. campus dining or bus services). Exposing this information to an adversary risks serious financial losses. Participants were shocked to find out this risk and told us that they even kept their credit cards in the smartphone card holder. Participants were also concerned after learning that Leap Motion could capture their partial computer screen. P6 said that *”This will definitely be a problem when I use an online bank to check my account”*. All participants told us that they would use Leap Motion with caution in the future.

8.2 User Feedback on Aili's Privacy Protection

We also conducted a user study to collect their feedback on Aili's privacy protection. We invited participants from our Leap Motion study to use Aili and comment on privacy-related issues. We are also interested in examining if participants are interested in using Aili at home or work. During the study, we demonstrate to participants how Aili works and shows them blockage maps of different hand gestures. They then complete a short questionnaire using a 7-point continuous numeric scale (1 strongly disagree, 7 strongly agree).

Overall, participants find no privacy issue using Aili (6.8, SD = 0.3). A participant comments that *”it seems to be more viable to use Aili instead of Leap Motion, since it will just capture the gesture without any privacy concerns.” (P5).* They all see themselves using Aili at home or in workplaces as both an input device and a table lamp (6.2, SD = 0.4). A participant comments that *”If the price is acceptable, I want to buy it” (P2).*

9 AILI USAGE SCENARIOS

To illustrate Aili's potential, we discuss five applications to showcase Aili's usage scenarios. We have made a demo video available at **`https://youtu.be/Fl1vVc3UGLA`**.

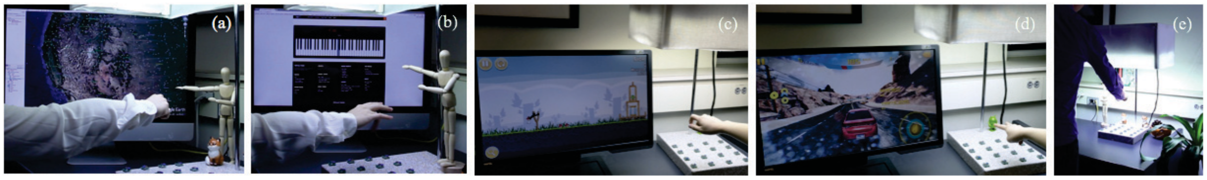


Fig. 14. Aili applications: (a) navigating Google Earth, (b) playing a piano game with a free hand in the light , (c) playing angry bird with pinch gestures, (d) playing a racing game with a ”V” gesture to trigger nitro booster, and (e) switching on/off light by gesturing under the lamp.

Manipulating Virtual 3D Objects Aili can replace a 3D mouse with complicated control buttons and simplify user's 3D control. Figure 14(a) shows an example where a user navigates Google Earth using Aili. Also, consider a user (e.g., mechanical engineer) sitting next to a table and editing virtual 3D objects in a software. Aili requires no extra input device and the lamp is always within the reach of the hand. While not implemented, other application scenarios include the following: a user can pinch the thumb and index finger to grab a virtual object and move the hand to translate the object in a 3D environment; opening the hand drops the object on the virtual floor.

Education and Gaming Aili can also facilitate user's learning of new skills or gaming. Figure 14(b) shows a user playing a virtual piano, where fingers can be mapped to a unique set of keys. Bending multiple fingers presses corresponding keys. Figure 14(c) demonstrates how a user leverages Aili to play Angry Bird with pinch gestures. Figure 14(d) illustrates a user performing a ”V” gesture to trigger booster in a racing game.

Controlling Home Appliances. Controlling the light at home requires the user to walk up to the switch. Although now a smartphone can be used as a remote controller, the task is still cumbersome as it requires the user to take out the phone, unlock it, navigate the app list, and open the controller app. With Aili, the user can freely gesture to switch on and off the light without leaving the desk or using the smartphone. For example, the user can use an open hand to turn on the light and a fist to turn it off (Figure 14(e)).

10 DISCUSSIONS

We now discuss the limitations of our current prototype and plans for future work.

Sensing Capability. We elaborate on the system's current sensing resolution both spatially and temporally, as well as its capability of handling palm rotation.

1) Spatial Resolution. Aili's spatial sensing resolution refers to the minimal horizontal finger movement that can be reconstructed by the system. It is bounded by the density of LEDs on the LED panel (i.e., the pixel interval of a blockage map). The system cannot recognize finger movement if its change on the blockage map is smaller than the LED/pixel interval (2.25 cm). This translates into 4.5-mm finger movement, assuming the hand is 10 cm above the lamp base. Increasing the LED density can improve the spatial sensing resolution (see Figure 7(a)). However, it can also degrade the robustness of blockage detection, as we described in the discussion of Figure 7(a). Photodiodes with higher sensing resolution can better detect small changes and potentially handle denser LEDs more robustly. We leave it to future exploration.

2) Temporal Resolution. Although Aili's pose reconstruction takes only 7.2 ms on average, the system's reconstruction rate is currently limited by the latency of acquiring blockage maps (25 ms). Thus, hand motion faster than 40 Hz leads to larger errors. However, this is not a fundamental limit of our approach, rather, it is an artifact of our use of Arduino Due boards for the ease of programming. Arduino Due has relatively low analog-to-digital converter (ADC) sampling rate and computation power, which results into 7 ms for sampling photodiode data and 15 ms for computing FFT to detect blockage. Furthermore, Arduino transmits data to a machine using a serial port. The port's low data rate (115 Kbps) adds 2-ms additional delay in data transfer. Nevertheless, these hardware constraints can be removed by using micro-controllers with faster ADC and communication interface. For example, the RM57L843 [20] micro-controller has 330-MHz CPU clock and 1.6-Msps ADC. With a communication interface such as USB 2.0 (480 Mbps), the delay of data transfer will be negligible. The resulting delay of acquiring blockage maps can be well controlled within 7 ms, allowing up to 140-Hz reconstruction rate.

3) Rotation. The current system requires a hand in the air (at least 5 cm above the table) and the forearm intersects the panel's long edge. The palm does not need to be strictly parallel to the lamp base, as the system supports a palm's roll angle up to ±45° and palm's pitch angle up to ±30°. Palm rotation within above range achieves 8° accuracy on average, otherwise its angular error becomes larger than 12° because finger occlusions are severer. Such occlusions can be addressed with more light rays from the sides. In future work, we plan to examine adding tilted LED panels at the lamp top, so that they emit light rays from more diverse directions to handle a wider range of palm rotation.

System Portability The current Aili prototype has limited portability since it requires LEDs and photodiodes at two sides and its LED panel is relatively large. To improve system portability, we will explore two aspects. First, we will study embedding photodiodes inside the LED panel to make Aili a standalone sensing panel. In this scenario photodiodes sense the light reflected by hand and the system leverages reflected light intensity to sense hand gestures. We will examine photodiodes with high sensitivity to deal with weak reflected light. Second, the LED panel can be made smaller, depending on the required sensing area of the application. Additionally, most electrical components (e.g., power supply, FPGAs boards) of the panel can be miniaturized. For examples, FPGAs can be replaced with AD9833 wave generators (9 mm^2 in size, 12.5 MHz). They can be hosted on a small PCB board integrated with a power supply, reducing the thickness of the LED panel to a few millimeters. It eases the integration of Aili to mobile devices (e.g., virtual reality headsets).

Broadening Sensing Scenarios Finally we plan to broaden Aili's sensing scenarios. We will study two-hand or even multi-user scenarios to allow richer user input and support users on collaborative tasks. The main challenge is that hands can block one another with overlapping blockage maps, which significantly increases reconstruction complexity and computation overhead. We will seek solutions to tracking individual hands. We will also extend Aili's ability to recognizing general objects (e.g., cups, phones) based on how they block light rays. It is challenging for objects that are fully or partially transparent. A possible solution is to leverage the raw frequency power after FFT computation to gauge the light penetration and infer object transparency.

11 CONCLUSION

We proposed a lightweight approach to reconstructing hand poses using only *binary* blockage information. We presented the design, development, and evaluation of Aili, a table lamp that senses how our hand blocks light rays to reconstruct arbitrary hand poses, without the need of cameras or on-body sensors. We evaluated Aili's usability and system performance via prototype experiments and user studies.

REFERENCES

[1] All of the Smart TV Gestures, Samsung Smart TV. **http://www.samsung.com/ph/smarttv/common/guide_book_3p_si/main.html**. (????).

[2] Nest Protect smoke + CO alarm. **https://store.nest.com/product/smoke-co-alarm**. (????).

[3] Nova Lighting 1010604. **http://www.totallyfurniture.com/nova-1010604.html**. (????).

[4] 2014. Hand Physics Controller. **https://www.assetstore.unity3d.com/en/#!/content/21105**. (2014).

[5] 2015. Google Soli Project. **https://www.google.com/atap/project-soli/**. (2015).

[6] 2016. Leap Motion Visualizer. **https://www.leapmotion.com/setup**. (2016).

[7] 2016a. LIGHTING DESIGN. **http://www.bristolite.com/interfaces/media/Footcandle%20Recommendations%20by%20Guth.pdf**. (2016).

[8] 2016b. Recommended Light Levels. **https://www.noao.edu/education/QLTkit/ACTIVITY_Documents/Safety/LightLevels_outdoor+indoor.pdf**. (2016).

[9] Vassilis Athitsos and Stan Sclaroff. 2003. Estimating 3D hand pose from a cluttered image. In *Proc. of CVPR.*

[10] Luca Ballan, Aparna Taneja, Jürgen Gall, Luc Van Gool, and Marc Pollefeys. 2012. Motion capture of hands in action using discriminative salient points. In *Computer Vision–ECCV 2012*. Springer, 640–653.

[11] Jeff C Becker and Nithish V Thakor. 1988. A study of the range of motion of human fingers with application to anthropomorphic designs. *Biomedical Engineering, IEEE Transactions on* 35, 2 (1988), 110–117.

[12] Paul Bratley, Bennett L Fox, and Harald Niederreiter. 1992. Implementation and tests of low-discrepancy sequences. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 2, 3 (1992), 195–213.

[13] Matthieu Bray, Esther Koller-Meier, and Luc Van Gool. 2004. Smart particle filtering for 3D hand tracking. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on.* IEEE, 675–680.

[14] Liwei Chan, Yi-Ling Chen, Chi-Hao Hsieh, Rong-Hao Liang, and Bing-Yu Chen. 2015. CyclopsRing: Enabling Whole-Hand and Context-Aware Interactions Through a Fisheye Ring. In *Proc. of UIST.*

[15] Martin de La Gorce, David J Fleet, and Nikos Paragios. 2011. Model-based 3d hand pose estimation from monocular video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 33, 9 (2011), 1793–1805.

[16] Artem Dementyev and Joseph A Paradiso. 2014. WristFlex: Low-power gesture input with wrist-worn pressure sensors. In *Proc. of UIST.*

[17] Jeremy Gummeson, Bodhi Priyantha, and Jie Liu. 2014. An Energy Harvesting Wearable Ring Platform for Gesture input on Surfaces. In *Proc. of MobiSys.*

[18] Sidhant Gupta, Daniel Morris, Shwetak Patel, and Desney Tan. 2012. SoundWave: Using the Doppler Effect to Sense Gestures. In *Proc. of CHI.*

[19] Roberto Hoyle, Robert Templeman, Steven Armes, Denise Anthony, David Crandall, and Apu Kapadia. 2014. Privacy behaviors of lifeloggers using wearable cameras. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing.* ACM, 571–582.

[20] Texas Instruments. RM57L843 16/32-Bit RISC Flash Microcontroller. **http://www.ti.com/product/RM57L843**. (????).

[21] Stephen Joe and Frances Y Kuo. 2003. Remark on algorithm 659: Implementing Sobol's quasirandom sequence generator. *ACM Transactions on Mathematical Software (TOMS)* 29, 1 (2003), 49–57.

[22] Stephen Joe and Frances Y Kuo. 2008. Constructing Sobol sequences with better two-dimensional projections. *SIAM Journal on Scientific Computing* 30, 5 (2008), 2635–2654.

[23] Derek G Kamper, T George Hornby, and William Z Rymer. 2002. Extrinsic flexor muscles generate concurrent flexion of all three finger joints. *Journal of biomechanics* 35, 12 (2002), 1581–1589.

[24] Bryce Kellogg, Vamsi Talla, and Shyamnath Gollakota. 2014. Bringing Gesture Recognition to All Devices. In *Proc. of NSDI*.

[25] Cem Keskin, Furkan Kıraç, Yunus Emre Kara, and Lale Akarun. 2012. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *Computer Vision–ECCV 2012*. Springer, 852–863.

[26] David Kim, Otmar Hilliges, Shahram Izadi, Alex D. Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. 2012. Digits: Freehand 3D Interactions Anywhere Using a Wrist-worn Gloveless Sensor. In *Proc. of UIST*.

[27] David Kim, Shahram Izadi, Jakub Dostal, Christoph Rhemann, Cem Keskin, Christopher Zach, Jamie Shotton, Timothy Large, Steven Bathiche, Matthias Nießner, and others. 2014. RetroDepth: 3D silhouette sensing for high-precision input on and above physical surfaces. In *Proc. of CHI*.

[28] Ye-Sheng Kuo, Pat Pannuto, Ko-Jen Hsiao, and Prabal Dutta. 2014. Luxapose: Indoor positioning with mobile phones and visible light. In *Proc. of MobiCom*.

[29] Hong Li, Wei Yang, Jianxin Wang, Yang Xu, and Liusheng Huang. 2016b. WiFinger: talk to your smart devices with finger-grained gesture. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 250–261.

[30] Liqun Li, Pan Hu, Chunyi Peng, Guobin Shen, and Feng Zhao. 2014. Epsilon: A Visible Light Based Positioning System. In *Proc. of NSDI*.

[31] Tianxing Li, Chuankai An, Zhao Tian, Andrew T Campbell, and Xia Zhou. 2015. Human sensing using visible light communication. In *Proc. of MobiCom*.

[32] Tianxing Li, Qiang Liu, and Xia Zhou. 2016a. Practical human sensing in the light. In *Proc. of MobiSys*.

[33] Jess McIntosh, Charlie McNeill, Mike Fraser, Frederic Kerber, Markus Löchtefeld, and Antonio Krüger. 2016. EMPress: Practical Hand Gesture Classification with Wrist-Mounted EMG and Pressure Sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2332–2342.

[34] Jon Moeller and Andruid Kerne. 2012. ZeroTouch: An Optical Multi-touch and Free-air Interaction Architecture. In *Proc. of CHI*.

[35] William J Morokoff and Russel E Caflisch. 1994. Quasi-random sequences and their discrepancies. *SIAM Journal on Scientific Computing* 15, 6 (1994), 1251–1279.

[36] Harald Niederreiter. 1988. Low-discrepancy and low-dispersion sequences. *Journal of number theory* 30, 1 (1988), 51–70.

[37] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. 2011. Efficient model-based 3D tracking of hand articulations using Kinect.. In *BmVC*, Vol. 1. 3.

[38] Iason Oikonomidis, Manolis IA Lourakis, and Antonis A Argyros. 2014. Evolutionary quasi-random search for hand articulations tracking. In *Proc. of CVPR*.

[39] Santiago Ortega-Avila, Bogdana Rakova, Sajid Sadi, and Pranav Mistry. 2015. Non-invasive optical detection of hand gestures. In *Proceedings of the 6th Augmented Human International Conference*. ACM, 179–180.

[40] William H Press, Saul A Teukolsky, William T Vetterling, and Brian P Flannery. 2007. Numerical recipes: the art of scientific computing. (2007).

[41] Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. 2014. Realtime and robust hand tracking from depth. In *Proc. of CVPR*.

[42] Jun Rekimoto. 2001. Gesturewrist and gesturepad: Unobtrusive wearable interaction devices. In *Wearable Computers, 2001. Proceedings. Fifth International Symposium on*. IEEE, 21–27.

[43] Wenjie Ruan, Quan Z Sheng, Lei Yang, Tao Gu, Peipei Xu, and Longfei Shangguan. 2016. AudioGest: enabling fine-grained hand gesture detection by decoding echo signal. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 474–485.

[44] Dragos Sbîrlea, Michael G Burke, Salvatore Guarnieri, Marco Pistoia, and Vivek Sarkar. 2013. Automatic detection of inter-application permission leaks in Android applications. *IBM Journal of Research and Development* 57, 6 (2013), 10–1.

[45] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, Daniel Freedman, Pushmeet Kohli, Eyal Krupka, Andrew Fitzgibbon, and Shahram Izadi. 2015. Accurate, Robust, and Flexible Real-time Hand Tracking. In *Proc. of CHI*.

[46] Il'ya Meerovich Sobol'. 1967. On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki* 7, 4 (1967), 784–802.

[47] Srinath Sridhar, Antti Oulasvirta, and Christian Theobalt. 2013. Interactive markerless articulated hand motion tracking using RGB and depth data. In *Proc. of CVPR*.

[48] Bjoern Stenger, Paulo RS Mendonça, and Roberto Cipolla. 2001. Model-based 3D tracking of an articulated hand. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, Vol. 2. IEEE, II–310.

[49] Paul Strohmeier, Roel Vertegaal, and Audrey Girouard. 2012. With a flick of the wrist: stretch sensors as lightweight input for mobile devices. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction.* ACM, 307–308.

[50] Li Sun, Souvik Sen, Dimitrios Koutsonikolas, and Kyu-Han Kim. 2015. Widraw: Enabling hands-free drawing in the air on commodity wifi devices. In *Proc. of MobiCom.*

[51] Andrea Tagliasacchi, Matthias Schröder, Anastasia Tkach, Sofien Bouaziz, Mario Botsch, and Mark Pauly. 2015. Robust Articulated-ICP for Real-Time Hand Tracking. In *Computer Graphics Forum*, Vol. 34. Wiley Online Library, 101–114.

[52] Danhang Tang, Hyung Chang, Alykhan Tejani, and Tae-Kyun Kim. 2014. Latent regression forest: Structured estimation of 3d articulated hand posture. In *Proc. of CVPR.*

[53] Danhang Tang, Tsz-Ho Yu, and Tae-Kyun Kim. 2013. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *Proc. of CVPR.*

[54] Jonathan Taylor, Lucas Bordeaux, Thomas Cashman, Bob Corish, Cem Keskin, Toby Sharp, Eduardo Soto, David Sweeney, Julien Valentin, Benjamin Luff, and others. 2016. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 143.

[55] Robert Templeman, Zahid Rahman, David Crandall, and Apu Kapadia. 2012. PlaceRaider: Virtual theft in physical spaces with smartphones. *arXiv preprint arXiv:1209.5982* (2012).

[56] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. 2014. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (TOG)* 33, 5 (2014), 169.

[57] Jue Wang, Deepak Vasisht, and Dina Katabi. 2014. RF-IDraw: Virtual Touch Screen in the Air Using RF Signals. In *Proc. of SIGCOMM.*

[58] Rui Wang, Fanglin Chen, Zhenyu Chen, Tianxing Li, Gabriella Harari, Stefanie Tignor, Xia Zhou, Dror Ben-Zeev, and Andrew T Campbell. 2014. StudentLife: assessing mental health, academic performance and behavioral trends of college students using smartphones. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing.* ACM, 3–14.

[59] Robert Wang, Sylvain Paris, and Jovan Popović. 2011. 6D Hands: Markerless Hand-tracking for Computer Aided Design. In *Proc. of UIST.*

[60] Saiwen Wang, Jie Song, Jamie Lien, Ivan Poupyrev, and Otmar Hilliges. 2016. Interacting with Soli: Exploring Fine-Grained Dynamic Gesture Recognition in the Radio-Frequency Spectrum. In *Proc. of UIST.*

[61] Yangang Wang, Jianyuan Min, Jianjie Zhang, Yebin Liu, Feng Xu, Qionghai Dai, and Jinxiang Chai. 2013. Video-based hand manipulation capture through composite motion control. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 43.

[62] Zachary Weinberg, Eric Y Chen, Pavithra Ramesh Jayaraman, and Collin Jackson. 2011. I still know what you visited last summer: Leaking browsing history via user interaction and side channel attacks. In *Security and Privacy (SP), 2011 IEEE Symposium on.* IEEE, 147–161.

[63] Mark Weiser. 1999. The Computer for the 21st Century. *SIGMOBILE Mob. Comput. Commun. Rev.* 3, 3 (July 1999), 3–11.

[64] Ying Wu, John Y Lin, and Thomas S Huang. 2001. Capturing natural hand articulation. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, Vol. 2. 426–432.

[65] Chao Xu, Parth H. Pathak, and Prasant Mohapatra. 2015. Finger-writing with Smartwatch: A Case for Finger and Hand Gesture Recognition Using Smartwatch. In *Proc. of HotMobile.*

[66] Nan Xu, Fan Zhang, Yisha Luo, Weijia Jia, Dong Xuan, and Jin Teng. 2009. Stealthy video capturer: a new video-based spyware in 3g smartphones. In *Proceedings of the second ACM conference on Wireless network security.* ACM, 69–78.

[67] Hui-Shyong Yeo, Gergely Flamich, Patrick Schrempf, David Harris-Birtill, and Aaron Quigley. 2016. RadarCat : Radar Categorization for input and interaction. In *Proc. of UIST.*

[68] Yang Zhang and Chris Harrison. 2015. Tomo: Wearable, Low-Cost Electrical Impedance Tomography for Hand Gesture Recognition. In *Proc. of UIST.*

[69] Chen Zhao, Ke-Yu Chen, Md Tanvir Islam Aumi, Shwetak Patel, and Matthew S. Reynolds. 2014. SideSwipe: Detecting In-air Gestures Around Mobile Devices Using Actual GSM Signal. In *Proc. of UIST.*

[70] Wenping Zhao, Jinxiang Chai, and Ying-Qing Xu. 2012. Combining marker-based mocap and RGB-D camera for acquiring high-fidelity hand motion data. In *Proceedings of the ACM SIGGRAPH/eurographics symposium on computer animation.* Eurographics Association, 33–42.