Human Sensing Using Visible Light Communication

Tianxing Li, Chuankai An, Zhao Tian, Andrew T. Campbell, and Xia Zhou
Department of Computer Science, Dartmouth College, Hanover, NH
{tianxing, chuankai, tianzhao, campbell, xia}@cs.dartmouth.edu

ABSTRACT

We present LiSense, the first-of-its-kind system that enables both data communication and fine-grained, real-time human skeleton reconstruction using Visible Light Communication (VLC). LiSense uses shadows created by the human body from blocked light and reconstructs 3D human skeleton postures in real time. We overcome two key challenges to realize shadow-based human sensing. First, multiple lights on the ceiling lead to diminished and complex shadow patterns on the floor. We design light beacons enabled by VLC to separate light rays from different light sources and recover the shadow pattern cast by each individual light. Second, we design an efficient inference algorithm to reconstruct user postures using 2D shadow information with a limited resolution collected by photodiodes embedded in the floor. We build a 3 m $\times$ 3 m LiSense testbed using off-the-shelf LEDs and photodiodes. Experiments show that LiSense reconstructs the 3D user skeleton at 60 Hz in real time with 10° mean angular error for five body joints.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication

Keywords

Visible light communication; sensing; skeleton reconstruction

1. INTRODUCTION

Light plays a multifaceted role (e.g., illumination, energy source) in our life. Advances on Visible Light Communication (VLC) [30, 59] add a new dimension to the list: data communication. VLC encodes data into light intensity changes at a high frequency imperceptible to human eyes. Unlike conventional RF radio systems that require complex signal processing, VLC uses low-cost, energy-efficient Light Emitting Diodes (LEDs) to transmit data. Any devices equipped with light sensors (photodiodes) can recover data by monitoring light changes. VLC has a number of appealing properties. It reuses existing lighting infrastructure, operates on an unregulated spectrum band with bandwidth 10K times greater than the RF spectrum, and importantly, is secure (i.e., does not penetrate walls, resisting eavesdropping), energy-efficient, and free of electromagnetic interference.

MobiCom'15, September 7–11, 2015, Paris, France.

DOI: http://dx.doi.org/10.1145/2789168.2790110.

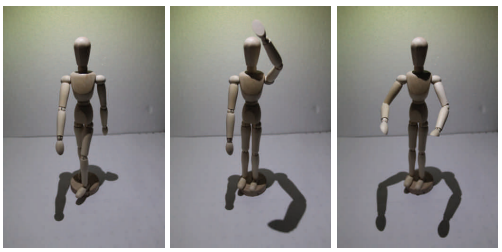


Figure 1: Shadow cast by varying manikin postures under an LED light (CREE XM-L). In this scaled-down table-top testbed, the manikin is 33 cm in height, and the LED light is 22 cm above the manikin.

In this paper, we push the envelope further and ask: *Can light turn into a ubiquitous sensing medium that tracks what we do and senses how we behave?* Envision a smart space (e.g., home, office, gym) that takes the advantage of the ubiquity of light as a medium that integrates data communication and human sensing [71]. Smart devices (e.g., smart glasses, smart watches, smartphones) equipped with photodiodes communicate using VLC. More importantly, light also serves as a passive sensing medium. Users can continuously gesture and interact with appliances and objects in a room (e.g., a wall mounted display, computers, doors, windows, coffee machine), similar to using the Kinect [57] or Wii in front of a TV, but there are no cameras (high-fidelity sensors with privacy concerns) monitoring users, neither any on-body devices or sensors that users have to constantly wear or carry [27, 62], just LED lights on the ceiling and photodiodes on the floor.

The key idea driving light sensing is strikingly simple: shadows. Any opaque object (e.g., human body) obstructs a light beam, resulting in a silhouette behind the object. Because the wavelength of visible light is measured in nanometers, any macroscopic object can completely block the light beam, much more significant than the radio frequencies [16, 50, 65, 70]. The shadow cast on the floor is essentially a two-dimensional projection of the 3D object. As the object moves and changes its shape, different light beams are blocked and the projected shadow changes at light speed – the same principle as the shadow puppet. Thus, by analyzing a continuous stream of shadows cast on the floor, we can infer a user's posture and track her behavior. As a simple illustration, Figure 1 shows the shadow shapes for varying manikin postures under a single light.

We present *LiSense, the first-of-its-kind system that enables fine-grained, real-time user skeleton reconstruction*[1] *at a high frame rate (60 Hz) using visible light communication*. LiSense consists

[1]We define skeleton reconstruction as calculating the vectors of the skeleton body segments in the 3D space.

of VLC-enabled LED lights on the ceiling and low-cost photodiodes on the floor.[2] LiSense aggregates the light intensity data from photodiodes, recovers the shadow cast by individual LED light, and continuously reconstructs a user's skeleton posture in real time. LiSense's ability of performing 3D skeleton reconstruction in real time puts little constraints on the range of gestures and behaviors that LiSense can sense, which sets a key departure from existing work that either targets a limited set of gestures [3, 17, 46] or only tracks user's 2D movements [2, 4, 10]. More importantly, by integrating both data communication and human sensing into the ubiquitous light, LiSense also fundamentally differs from vision-based skeleton tracking systems (e.g., Kinect) that are built solely for the sensing purpose. In addition, these systems rely on cameras to capture high-resolution video frames, which bring privacy concerns as the raw camera data can be leaked to the adversary [52, 64]. While prior vision methods [55, 56] have leveraged shadow to infer human gestures, they work strictly under a single light source and do not apply in a natural indoor setting with multiple light sources.

LiSense overcomes two key challenges to realize shadow-based light sensing: 1) *Shadow Acquisition:* Acquiring shadows using low-cost photodiodes is challenging in practice. In the presence of multiple light sources, light rays from different directions cast a diluted composite shadow, which is more complex than a shadow cast by a single light source. A shadow can also be greatly influenced by ambient light (e.g., sunlight). Both factors limit the ability of photodiodes detecting the light intensity drop inside a shadow. To address this challenge, LiSense leverages the fact that each light is an active transmitter using VLC and designs *light beacons* to separate light rays from individual LEDs and ambient light. Each LED emits light beacons by transmitting (i.e., flashing) at a unique frequency. LiSense transforms the light intensity perceived by each photodiode over time to the frequency domain. By monitoring frequency power changes, LiSense detects whether the photodiode is suddenly blocked from an LED and aggregates the detection results from all photodiodes to recover the *shadow map* cast by each light.

2) *Shadow-based Skeleton Reconstruction:* Shadow maps measured by photodiodes are 2D projections with a limited resolution (constrained by the photodiode density). Such low-resolution, imperfect shadow images pose significant challenges to reconstruct a user's 3D skeleton. Existing computer vision algorithms [11, 19, 21, 24, 42, 66, 51] cannot be directly applied to this problem because they all deal with video frames in a higher resolution and are often augmented with the depth information. LiSense overcomes this challenge by combining shadows cast by light sources in different directions and infers the 3D vectors of key body segments that best match shadow maps. LiSense fine-tunes the inferences using a Kalman filter to take into account movement continuity and to further reduce the skeleton reconstruction errors.

LiSense Testbed. We build a 3 m $\times$ 3 m LiSense testbed (Figure 9), using five commercial LED lights, 324 low-cost, off-the-shelf photodiodes, 29 micro-controllers, and a server. We implement light beacons by programming the micro-controllers that modulate LEDs. We implement blockage detection and 3D skeleton reconstruction algorithms on the server, which generates a stream of shadow maps and continuously tracks user gestures. The reconstruction results are visualized in real time using an animated user skeleton (Figure 11). We test our system with 20 gestures and seven users in diverse settings. Our key findings are as follows:

- LiSense reconstructs a user's 3D skeleton with the average angular error of 10° for five key body joints;
- LiSense generates shadow maps in real time. It is able to produce shadow maps of all LEDs every 11.8 ms, reaching the same level of capturing video frames yet without using cameras;
- LiSense tracks user gestures in real time. It reconstructs the user skeleton within 16 ms based on five shadow maps, thus generating 60 reconstructed postures, each of which consists of the 3D vectors of five key body segments. The reconstructed skeleton is displayed in real time (60 FPS), similar to playing a video at a high frame rate;
- LiSense is robust in diverse ambient light settings (morning, noon, and night) and users with different body sizes and shapes.

Contributions. We make the following contributions:

- We propose for the first time the concept of continuous user skeleton reconstruction based on visible light communication, which enables light to be a medium for both communication and passive human sensing;
- We design algorithms to extract the shadow of each individual light source and reconstruct 3D human skeleton posture continuously using only a stream of low-resolution shadow information;
- We build the first testbed implementing real-time, human skeleton reconstruction based on VLC, using off-the-shelf, low-cost LEDs, photodiodes, and micro-controllers in an indoor environment;
- Using our testbed, we test our system with diverse gestures and demonstrate that it can reconstruct a user skeleton continuously in real time with small reconstruction angular errors.

Our work takes the first step to go beyond conventional radio spectrum and demonstrates the potential of using visible light spectrum for both communication and fine-grained human sensing. We believe that with its unbounded bandwidth, light holds great potential to mitigate the spectrum crunch crisis. By expanding the applications VLC can enable, we hope that our work can trigger new radical thinkings on VLC applications. Our work examines the interplay between wireless networking, computer vision, and HCI, opening the gate to new paradigms of user interaction designs.

2. LIGHT SHADOW EFFECT

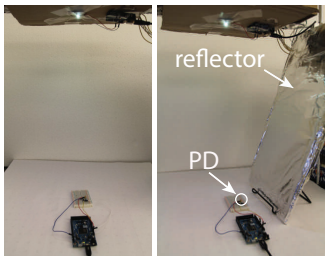
Shadow is a common phenomenon we observe everyday. It is easily recognizable under a single light source by unaided human eyes. Our goal is to understand whether off-the-shelf, low-cost photodiodes can reliably detect the light intensity drop in the shadow. If so, we can deploy them on the floor and aggregate their light intensity data to obtain the shadow cast by a human body. In this section, we first study the impact of a blocking object on light propagation using low-cost photodiodes. We then examine the challenges of shadow-based analysis in the presence of multiple lights.

2.1 Experiments on Blocking the Light

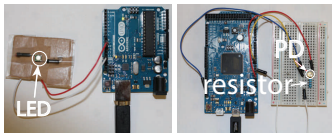
Consider a single photodiode on the floor, we hypothesize that if any opaque object stands in the direct path between the point light source and the photodiode, the photodiode will not be able to perceive any light coming from this point light source. Thus, the photodiode observes a light intensity drop compared to the case when there is no object blocking its direct path to the light source.

To confirm our hypothesis, we build a scaled-down table-top testbed (Figure 2) using commercial LED lights (CREE XM-L) and low-cost photodiodes (Honeywell SD3410-001). We set up a single LED chipset as the point light source at 55 cm height and place the photodiode directly below the light. By default we calibrate the photodiode's location using a plumb bob to ensure 0° of light's incidence angle. The photodiode has 90° field of vision (FoV), i.e., it

[2]Engineering photodiodes on the floor sounds labor-intensive today, but it can be eased by smart fabric [1, 45] (see more in § 7).

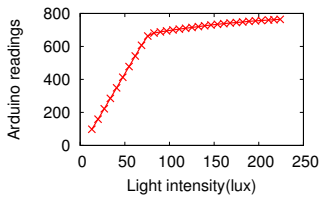


(a) Default setup (left) and setup w/ reflector (right)

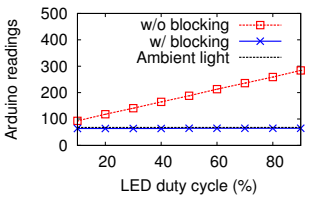


(b) LED and photodiode (PD) w/ Arduino boards

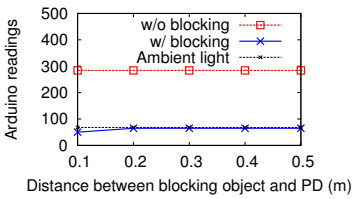
Figure 2: Experiment setup with an LED and a photodiode (a), both attached to micro-controllers (b).



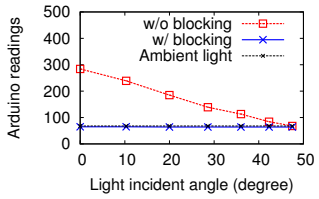
(a) Light intensity vs Arduino reading



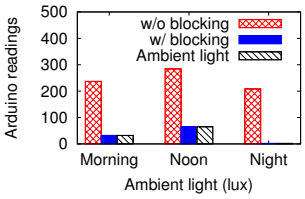
(b) Light source light intensity



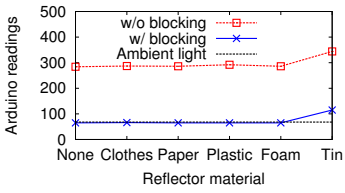
(c) Blocking object distance



(d) Incident angle



(e) Ambient light



(f) Multi-path

Figure 3: Experiments on blocking the light using our scaled-down table-top testbed. (a) shows that the measured Arduino reading is directly proportional to the perceived light intensity, where the slope decreases after the photodiode enters its saturation range. (b)-(f) show the impact of blockage on the measured Arduino reading under varying settings.

can sense incoming light with incidence angle within 45°. To fetch signal continuously from the photodiode, we cascade the photodiode and a resistor (10 KΩ) and measure the resistor voltage using a micro-controller (Arduino DUE in Figure 2(b)). It maps the measured voltage to an integer between 0 and 1023. Since the photodiode's output current is directly proportional to the perceived light intensity, resistor voltage (thus the Arduino reading) reflects the perceived light intensity. Using a light meter (EXTECH 401036) we have verified that the Arduino reading is directly proportional to the perceived light intensity (Figure 3(a)). To understand the impact of blockage, we place a 10 cm × 10 cm × 2 cm wood plate between the photodiode and the LED, and compare the Arduino readings before and after placing the wood plate[3]. We aim to answer the following key questions:

Q1: Is the shadow dependent on the light intensity of the light source? We first examine how the light source brightness affects the photodiode's sensing data upon blockage. We connect the LED to an Arduino UNO board to vary the LED's duty cycle from 10% to 90% (Figure 2(b)), resulting in light intensities from 5 to 30 lux perceived by the photodiode. For a given duty cycle, we record the average Arduino reading before and after blocking the LED and plot the results in Figure 3(b). We observe that upon blockage, the Arduino's reading reports only the ambient light in all duty cycle settings, meaning that an opaque object completely blocks the light rays regardless of the brightness of the light source.

Q2: Does the distance between the blocking object and the light source matter? Next, we test whether the relative distance between the blocking object and the light source affects the photodiode's sensed light intensity. To do so, we fix the LED's duty cycle to 90%, move the wood plate along the line between the LED and the photodiode, and record the Arduino data at each distance. Figure 3(c) shows that as long as the object stays in the direct path between the LED and the photodiode, the light beam is completely blocked regardless of the relative distance of the blocking object.

Q3: How does the light incidence angle come into play? Because a photodiode has a limited viewing angle and can perceive incoming light only within its FoV, we further examine whether it can detect blockage under varying light incidence angles. We move the photodiode horizontally with 10-cm intervals and record the Arduino's reading before and after blockage. As expected, the perceived light intensity gradually drops as the photodiode moves further away from the LED (Figure 3(d)). More importantly, at all locations (incidence angles), the light beam blockage result in a significant drop in the Arduino's reading. The drop is less significant when the incidence angle approaches half of the photodiode's FoV. This is because the photodiode can barely sense any light coming at its FoV and thus blocking the light beam has a negligible impact.

Q4: What is the impact of ambient light? We also perform our measurements during different time of a day as the ambient light varies. In Figure 3(e), we plot the Arduino reading before and after blockage as the ambient light intensity increases from 2 to 100 lux. In all conditions, we observe a significant drop in the Arduino reading. Because the photodiode senses a combination of the ambient light and the light from the LED, its perceived light intensity increases as the ambient light intensity increases.

Q5: How significant is the light multi-path effect? Would it diminish the shadow? Visible light is diffusive in nature. While a object blocks the direct path between the photodiode and the LED, light rays can bounce off surrounding objects and reach the photodiode from multiple directions. Since the photodiode perceives a combination of light rays coming in all directions, this multi-path effect can potentially reduce the light intensity drop caused by blocking the direct path. To examine the impact of the multi-path effect, we place a flat board vertically close to the LED to increase the reflected light rays (Figure 2(a), right) and record the Arduino's reading with and without blocking the direct path to the LED. Among all types of material we have tested, the significant drop in the Arduino's reading is consistent (Figure 3(f)). Thus, light in the direct path dominates the perceived light intensity. The tin has the highest light intensity because of its minimal energy loss during reflection.

Overall, our experiment results confirm that opaque objects can effectively block light in diverse settings and the blockage can be detected by low-cost photodiodes under a single point light source.

[3]We also measured the blockage impact using different body parts (e.g., arms, hands) of a manikin and observed similar results.

(a) 2 LEDs (b) 3 LEDs (c) 4 LEDs (d) 5 LEDs (e) Setup w/ PD (f) Sensed light intensity

Figure 4: Shadow cast by multiple LEDs on the table-top testbed (a)-(d). We further measure the light intensity change caused by blockage using off-the-shelf photodiode (e). Light intensity drop caused by blockage is less significant under more LEDs (f).

2.2 Where is My Shadow?

Detecting a shadow is relatively straightforward under a single point light source. However, when there are multiple light sources present, shadow detection becomes much more challenging. This is because light rays from different sources result in a *composite shadow*, which comprises shadow components created and fused by multiple light sources. Figure 4 illustrates the resulting shadow of the manikin as we switch on more LED lights in our table-top testbed. We make two key observations. *First*, the shadow is gradually diluted as more LEDs are switched on. This is because there are more light rays coming from different directions, and hence blocking a beam from one LED does not necessarily block the light beams from other LEDs, leaving a fading shadow. *Second*, the shadow shape becomes more complex under more LEDs as a result of superimposing different shadows cast by individual LEDs. As a result, it becomes harder to infer the manikin's posture based upon the resulting tangled shape pattern.

While visually less noticeable, a shadow is also much harder to detect under these conditions using off-the-shelf photodiodes. In our experiments, we place the photodiode in a shadow region caused by the manikin's posture, gradually switch on more LEDs, and compare the Arduino's reading with and without the manikin's blockage (Figure 4(e)). We observe that as more LEDs are switched on, more light rays coming in different directions hit the shadow region and thus the perceived light intensity level rises. Furthermore, once three or more LED lights are switched on, the photodiode enters the saturation region (Figure 3(a)), thus blocking light rays from a single LED has a negligible impact on the Arduino reading. As a result, detecting shadow using these low-cost photodiodes is very challenging in practice under multiple lights.

In the next two sections, we introduce LiSense, which disambiguates composite shadows using VLC and continuously tracks user posture in real time.

3. DISAMBIGUATING SHADOWS

To disambiguate composite shadows created by multiple lights, LiSense recovers the shadow shape, referred to as the *shadow map*, resulting from each individual light source. Specifically, the shadow map associated with light source L_k is the shadow if only light source L_k is present. We describe this as disambiguating a composite shadow. The key challenge is that each photodiode perceives a combination of light rays coming from different light sources and cannot separate light rays purely based on the perceived light intensity (Figure 5(a)(b)).

To overcome this technical barrier, we leverage the fact that each LED light is an active transmitter using VLC. We instrument each light source to emit a unique *light beacon*, implemented by modulating the light intensity changes at a given frequency. By assigning a different frequency to each light source, we enable photodiodes to differentiate lights from different sources. This allows LiSense to recover the shadow cast by each individual light.

In this section, we first describe our design of light beacons, followed by the mechanism to detect blockage (shadow) and infer shadow maps.

3.1 Separating Light Using Light Beacons

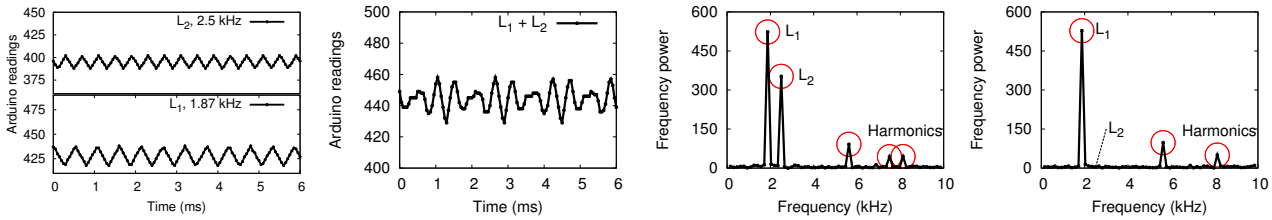
The design of light beacons is driven by the observation that while the perceived light intensity represents the sum of all incoming light rays, these light rays can be separated in the frequency domain if they flash at different frequencies. That is, if we transform a time series of perceived light intensity to the frequency domain using Fast Fourier Transform (FFT), we can observe frequency power peaks at the frequencies at which these light rays flash. Figure 5 shows an example with two LED lights flashing at 2.5 kHz and 1.87 kHz, respectively, in an unsynchronized manner. The light intensity perceived by the photodiode is a combination of these two light pulse waves, and yet FFT can decompose the light mixture and generate peaks at the two flashing frequencies. Thus, a light beacon can be implemented by programming each light source to flash at a unique frequency.

Benefits. Light beacons bring three key benefits when considering blockage detection. *First*, by examining the resulting frequency power peaks after applying FFT, we can separate light rays from different light sources. The frequency power at frequency f_i is approximately directly proportional to the intensity of light rays flashing at f_i. Thus, the changes in power peaks allow the photodiode to determine *which* lights are blocked. *Second*, light beacons also allow us to avoid interference from ambient light sources by applying a high pass filter (HPF). This is because the change of ambient light is random and generates frequency components close to zero in the frequency domain. *Third*, by separating light rays from different sources, we observe a much more significant drop in the frequency power caused by blocking a light, which is the key to achieving robust detection of blockage, especially when the photodiode perceives a weak light intensity because of a long distance or a large incidence angle.

Light Beacon Frequency Selection. Designing robust light beacons, however, is nontrivial, mainly because selecting the flashing frequency for each light source is challenging. Specifically, assume an LED flashes as a pulse wave with a duty cycle of D and flashing frequency of f, the Fourier series expansion of this pulse wave is:

$$f(t) = D + \sum_{n=1}^{\infty} \frac{2}{n\pi} \sin(\pi n D) \cos(2\pi n f t). \quad (1)$$

It indicates that the power emitted by the pulse wave is decomposed into the main frequency power, which is the first AC component when $n = 1$, and an infinite number of harmonics (components



(a) A single LED on (b) Two LEDs on (c) After applying FFT (d) After blocking L_2

Figure 5: Experiments with a photodiode (PD) and two LEDs (L_1 and L_2, 50% duty cycle) that flash at different frequencies. (a)-(b) show the PD's readings when only one LED is on and when both are on. PD perceives a combination of light rays, which however can be separated in the frequency domain after applying FFT (c). The frequency power of the flashing frequency f_i reflects the perceived intensity of light rays flashing at f_i. Thus the power peak at 2.5 kHz disappears after L_2 is blocked (d).

Algorithm 1: Selecting the candidates of flashing frequency for light beacons.

input : 1) R, signal sampling rate; 2) A, the number of FFT points; 3) $f_{flicker}$, the threshold to avoid flickering; 4) $f_{interval}$, the minimal interval between adjacent flashing frequencies
output: $f_{candidate}$, flashing frequency candidates for all LEDs

$f_{candidate} = \{\frac{R}{A} \times \lceil \frac{f_{flicker} \times A}{R} \rceil\}$
for $k \leftarrow \lceil \frac{f_{flicker} \times A}{R} \rceil + 1$ **to** $\frac{A}{2}$ **do**
  $f_k = \frac{R}{A} \times k$
  $valid = true$
  for $f_s \in f_{candidate}$ **do**
    if $(f_s \mod f_k) == 0)$ *OR* $(|f_s - f_k| \leq f_{interval})$ **then**
      $valid = false$
      break
  end
  if $valid$ **then** $f_{candidate} \leftarrow f_k \cup f_{candidate}$
end

with $n > 1$). Hence, an LED light L_i flashing at frequency f_i leads to not only a global power peak (main frequency power) at frequency f_i, but also small local power peaks at all the harmonics frequencies (Figure 5(c)). In other words, if the perceived light intensity from light L_i changes, it will affect not only the main frequency power at f, but also the power peaks at harmonics. To separate out the light rays and avoid interference across lights, we need to ensure that the harmonics do not overlap with the main frequencies of other lights. Tracking all harmonics is infeasible. In our design, we focus on the top-ten harmonics frequency components. This is because the harmonics frequency power drops significantly as n increases. We observe it becomes negligible once $n > 10$.

Furthermore, the flashing frequencies need to satisfy three additional constraints. *First*, since lights are also used for illumination, the flashing frequencies need to be above a threshold $f_{flicker}$ (1 kHz in our implementation) to avoid the flickering problem [32, 36, 48]. *Second*, the highest flashing frequency is limited by the sampling rate of the micro-controller fetching data from photodiodes. The Nyquist Shannon sampling theorem [25] says that it has to be no larger than $R/2$, where R is sampling rate. *Finally*, the adjacent frequencies have to be at least $f_{interval}$ away to ensure robust detection of frequency power peaks. We set $f_{interval} =$ 200 Hz based on a prior study [32]. Algorithm 1 details the procedure to select all candidate flashing frequencies satisfying all the above constraints.

We then assign the candidate flashing frequencies to all LED lights, such that the lights within each photodiode's viewing angle (FoV) flash at different frequencies. Since photodiodes have a limited FoV (90 degrees for Honeywell SD3410-001), each photodiode perceives only a small subset of all lights. Thus we do not need a large number of candidate frequencies to cover all lights and the system can easily scale up to more lights. Supporting denser lights requires more candidate flashing frequencies, which can be achieved by increasing the signal sampling rate R.

Light Beacon Overhead. Light beacons can be seamlessly integrated into existing VLC systems, enabling light to fulfill a dual role of data communication and human sensing. For VLC systems [5, 14, 36, 48] that use Frequency Shift Keying (FSK) to modulate data, all the data packets serve as light beacons, as long as LED lights within the FoV of a photodiode use different flashing frequencies to modulate data. For VLC systems that use other modulation schemes [34, 35], we can instrument each LED light to emit light beacons periodically, in the same way that Wi-Fi access points periodically transmit beacons. For an ADC sampling rate of 20 kHz and modulation window of 128 points, a light beacon lasting for 6.4 ms is sufficient for the photodiode to separate light rays. Thus, the overhead of transmitting light beacons is negligible given that a data packet typically lasts for hundreds of ms based on the IEEE 802.15.7 standard [8].

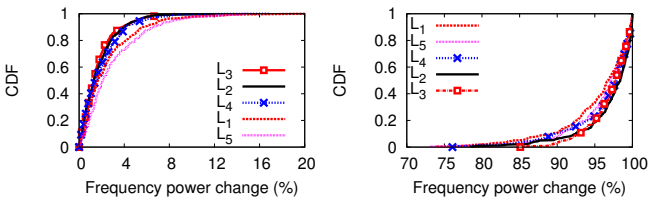
3.2 Blockage Detection

We detect blockage by transforming the time series of light intensity values of light beacons to the frequency domain and examining the frequency power changes. Specifically, the intensity of light rays from light L_i flashing at frequency f_i is represented by the frequency power of f_i. If an opaque object blocks the direct path from light L_i to a photodiode, the frequency power of f_i changes (Figure 5(d)).

To examine the impact of blockage on the frequency power, we mount five commercial LED lights (Cree CXA25) on an office ceiling (2.65 m in height), attach them to an Arduino UNO board, which modulates the Pulse Width Modulation (PWM) of each light to allow each LED to emit light beacons at a given frequency (Table 1 by running Algorithm 1). We place photodiodes (Figure 2(b)) at 324 locations in a 3 m x 3 m area on the floor. Each photodiode can perceive light rays from all LED lights. We then measure the readings of the Ardunio controllers connected to the photodiodes for 6.4 ms before and after blocking each LED light.

Figure 6 shows the CDF of relative frequency power change. Assume the $P_{ij}(t)$ is the frequency power of f_i (the flashing frequency of the light beacons from light L_i) at time t perceived by the photodiode at location p_j, its relative frequency power change $\Delta P_{ij}(t)$ is defined as:

$$\Delta P_{ij}(t) = |\frac{P_{ij}^{\text{nonBlock}} - P_{ij}(t)}{P_{ij}^{\text{nonBlock}}}|, \quad (2)$$



(a) w/o blocking (b) w/ blocking

Figure 6: Relative frequency power changes in the non-blocking (a) and blocking state (b).

where P_{ij}^{nonBlock} is the average of non-blocking frequency power from light L_i at location p_j. Clearly the frequency power change caused by blockage is much more significant (>70%) than that caused by the normal light intensity fluctuation[4] (<20%) in non-blocking state. Therefore, using a threshold τ (60% in our implementation) on the relative frequency power change can effectively detect the occurrence of blockage.

Shadow Map. By aggregating the blockage detection result from all photodiodes, we can recover the shadow map cast by each light L_i. Specifically, assuming N photodiodes on the floor, which can sense K LED lights within their FoVs, we define the shadow map $S_i(t)$ cast by LED light L_i at time t as: $S_i(t) = \{s_{ij}(t)|0 < j \leq N\}$, where $s_{ij}(t)$ indicates whether the direct path from location p_j to light L_i is blocked at time t, i.e., $s_{ij}(t) = 1$ if $\Delta P_{ij}(t) \geq \tau$, and $s_{ij}(t) = 0$ otherwise. Figure 8 shows five example shadow maps obtained from our testbed (§ 5) for a given user posture (Figure 8(a)). Since each shadow map is generated by blocking light rays from a different angle, the combination of the 2D shadow maps can be used to reconstruct the human skeleton in the 3D space.

4. FROM SHADOW MAPS TO POSTURE

Given the set of inferred shadow maps at time t, LiSense next reconstructs the user's 3D skeleton at t, referred to as the user's posture at time t. By continuously inferring the user's posture over time, LiSense recovers the user's gesture, which consists of a sequence of postures. While computer vision literature has studied similar problems on 3D human motion reconstruction, we face two new challenges here. *First*, constrained by the limited photodiode density, shadow maps have limited resolution (18 × 18 pixels in our testbed), far below the resolution of video frames (at least 640 × 480 pixels [40]) used by existing vision techniques [11, 21, 24, 66]. Also shadow maps are imperfect and a small portion of pixels can suffer from blockage detection errors. *Second*, a shadow map is only a two-dimensional projection of the 3D object and lacks the depth information. As shown in our prior experiment (Figure 3(c)), the blockage at a single photodiode is independent of its relative distance to the blocking object. Existing vision techniques [24, 19, 51], however, are typically augmented by the depth input.

To address these challenges, we combine shadow maps cast by LEDs from different perspectives to reconstruct the user's 3D skeleton. Inspired by existing methods on skeleton-based motion capture [20, 41, 58, 67], we design a shadow-based inference algorithm to compute the 3D vectors of the key user body segments. We also take into account the continuity of human movement and apply Kalman filter to iteratively fine-tune the current inference based on the prior inferred posture. Next we first describe the posture inference algorithm, followed by the fine-tuning using Kalman filter.

[4]The light intensity fluctuation in the non-blocking state is attributed to the ADC errors at the Arduino board, the imperfect pulse waves generated by the LED, and the photodiode noise.

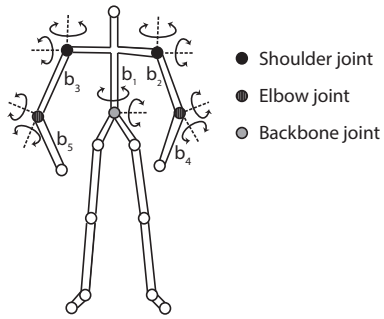


Figure 7: User skeleton model. We track key body segments b_i by rotating the key body joints. Each joint has two degrees of freedom because of human body's physical limit.

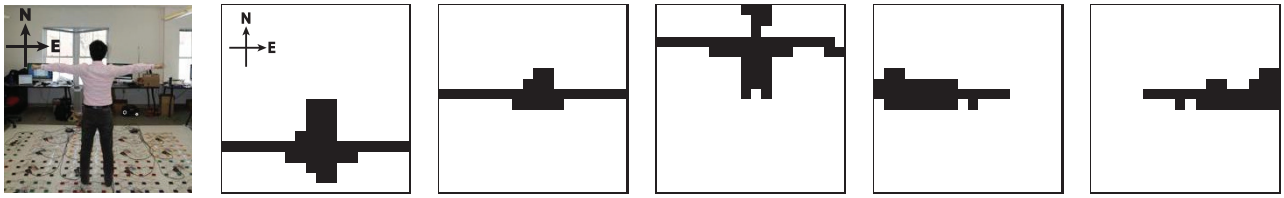
4.1 Inferring Posture based on Shadow

Our inference algorithm takes four inputs: 1) inferred shadow maps for all K LEDs ($S_1(t), ..., S_K(t)$) at time t, 2) 3D locations of all LED lights, 3) 3D locations of all N photodiodes, and 4) user's body parameters (e.g., body height, body part size). The algorithm does not require any training to reconstruct the user skeleton. It needs only initial calibration on user's body parameters.

User Skeleton Model. We model the user using a standard human skeleton model [20, 41, 58, 67] shown in Figure 7. We do not target specific pre-defined gestures and each body part can freely rotate under the physical constraints of the human body. We aim to reconstruct the set of 3D vectors B to represent M body segments, denoted by $B = \{b_i|0 < i \leq M\}$. In our implementation, we consider five body segments ($M = 5$): backbone, left and right upper arms, and left and right lower arms. We focus on upper-body segments because of the current testbed setting (see more in § 7). The inference algorithm, however, is applicable for all body segments. The movements of these body segments are controlled by the rotation of five key body joints: backbone joint, left and right shoulder joints, and left and right elbow joints, respectively. We assume that each key body joint has two degrees of freedom for its rotation (Figure 7) because of human body's physical limit. Then the rotation of each body joint can be represented by the actual rotation angles in both degrees. We use coordinate conversion, denoted by $g_m(\cdot)$, to transform a set of body joint rotation angles to the 3D vector of each body segment b_m.

Shadow-based Inference. At the core, our inference algorithm aims to seek the optimal 3D vectors for body segments, so that the resulting human skeleton best matches the shadow map cast by each LED. Leveraging the shadow information, our inference is based on a simple fact: If a photodiode at location p_j is blocked from LED L_i at time t, i.e., $s_{ij}(t) = 1$ in shadow map $S_i(t)$, then the light ray l_{ij} from L_i to p_j is blocked by at least one user body segment, meaning that the minimal perpendicular distance between light ray l_{ij} to all body segments should be smaller than the radius r_i of the body part that the corresponding body segment b_i represents. Therefore, we aggregate the blockage information from all shadow maps. We infer the optimal set of body segments $B^\star$ as the body segments that lead to the minimal summation of perpendicular distances between all blocked light rays and their closest body segments. In the meantime, the body segments should not block any light ray l_{ix} if photodiode p_x is not blocked from LED L_i, i.e., $s_{ix}(t) = 0$. Thus we represent the optimization problem as below:

$$B^\star = \underset{B \in \mathbb{B}'}{\text{argmin}} \sum_{\substack{s_{ij}(t)=1 \\ i \in [1,K], j \in [1,N]}} \min_{b_m \in B} (dist(l_{ij}, b_m) - r_m), \quad (3)$$



(a) User gesture (b) Inferred shadow maps (18x18 pixels) cast by five LEDs: L_1, L_2, L_3, L_4, and L_5 (Figure 9(a)) respectively.

Figure 8: Shadow maps obtained from LiSense testbed.

where $dist(l_1, l_2)$ calculates the perpendicular distance between two line segments l_1 and l_2, and $\mathbb{B}'$ contains the candidate 3D-vector set B satisfying the below constraint:

$$dist(l_{ix}, b_m) > r_m, \forall b_m \in B, \text{if } s_{ix}(t) = 0, x \in [1, N].$$

We iteratively optimize the objective function (Eq. (3)) for each body segment. The maximum number of iterations is 6 in our current implementation. Given the number of possible movements a body segment can make, the search space for the optimal $B^\star$ is daunting. We speed up the search in two ways. *First*, instead of examining all blocked photodiodes (shadow pixels), we check only the blocked photodiodes on the shadow boundary to reduce the number of blocked light rays needed to process. *Second*, we apply a greedy algorithm to prioritize the search order based on the body part size. We first consider the movement of the largest body part, which is the backbone. To do so, we rotate the backbone joint and search the optimal rotation angles of this joint to match the shadow maps. We then move on to the upper arms by rotating the shoulder joints, and finally the lower arms by rotating the elbow joints. At each step, we apply a greedy algorithm to identify the optimal rotation angles of each joint. In the end, we aggregate the optimal rotation angles of each joint, and apply $g_m(\cdot)$ to convert them to the optimal 3D vectors $B^\star$ of body segments.

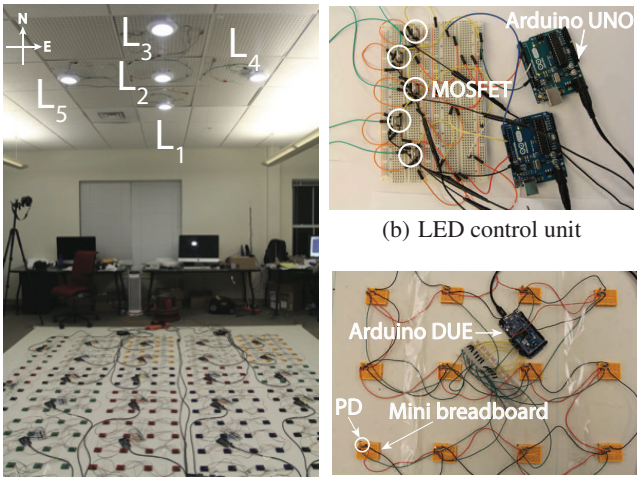
4.2 Fine-Tuning Posture Inference

We further fine-tune the inferred vectors of body segments $B^\star$, which can be affected by blockage detection errors (wrong shadow pixels) and the lack of information on body occlusion due to shadow map's low resolution and photodiode's limited viewing angle. The key idea is to leverage the prior inferred posture and smooth the inference result over time. A simple method is to average the rotation angle of each joint within a time window. However, determining the time window size is non-trivial. We observe that the final performance suffers from non-Gaussian noise and occlusion from other body parts.

To overcome the above problems, we apply a Kalman filter to smooth body part movements over time given the movement continuity. We model each body part movement as a stochastic process – it moves at a constant velocity with random perturbations in its trajectory. Let (x_t, y_t, z_t) denote the 3D position of a body segment's end point at time t and dx_t, dy_t, dz_t is the velocity in each dimension respectively, we represent the process as:

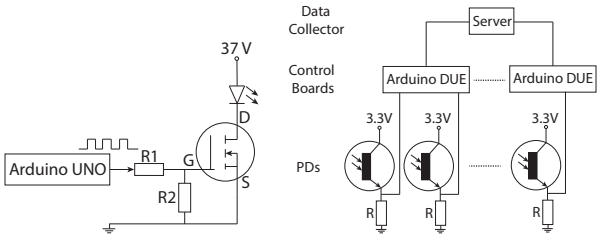
$$\begin{pmatrix} x_t \\ y_t \\ z_t \\ dx_t \\ dy_t \\ dz_t \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}}_{\text{Physical constrain matrix}} \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ z_{t-1} \\ dx_{t-1} \\ dy_{t-1} \\ dz_{t-1} \end{pmatrix} + \begin{pmatrix} n_x \\ n_y \\ n_z \\ n_{dx} \\ n_{dy} \\ n_{dz} \end{pmatrix},$$

where n_x, n_y, n_z are statistic variables representing the random perturbation in each dimension. Here the physical constraint matrix captures human body movement contiguity. The initial values of (x_t, y_t, z_t) and dx_t, dy_t, dz_t are known, and n_x, n_y, n_z are initialized to zeros. The Kalman filter then iteratively fine-tunes the body part position (x_t, y_t, z_t) and updates the statistic noise. We repeat the fine-tuning for each body segment. In the end, the fine-tuned body segments are visualized on the screen. By estimating the stochastic noise, Kalman filter further reduces the skeleton reconstruction errors and smooths the visualization.



(a) LiSense overview (b) LED control unit (c) 12 photodiodes + Arduino

Figure 9: LiSense testbed.



(a) LED control circuit (b) Circuit for photodiodes

Figure 10: The circuit designs for the LED control unit and for connecting photodiodes to micro-controllers.

5. LiSense TESTBED

We build LiSense using five off-the-shelf LED lights (CREE CXA25), 324 low-cost (<$2 in wholesale) photodiodes (Honeywell SD3410-001) in a 3 m × 3 m area, 29 micro-controllers (Arduino UNO and DUE), and a server (DELL M4600) in a research lab setting (Figure 9), where the ceiling height is 2.65 m. Each LED chipset is equipped with a commercial LED lampshade (CREE T67) and adjacent LED lights are 0.8 m away.

LED Lights. We mount five LED lights on the ceiling. We connect all five LED lights to two Arduino UNO boards using five out-

Table 1: LED flashing frequency.

LED light	L_1	L_2	L_3	L_4	L_5
PWM frequency (kHz)	2.5	1.9	2.2	3.4	1.6

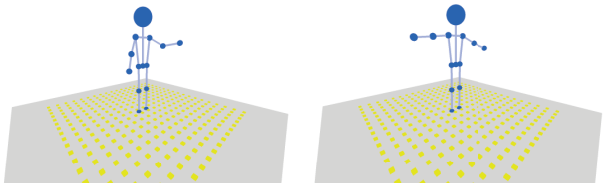


Figure 11: The visualized 3D skeleton reconstructed by LiSense for two example postures.

put pins and each output pin connects to a MOSFET driver module (Figure 10(a)). We fix the LED's duty cycle to 50%. We implement light beacons by programming the Arduino boards to modulate the PWM frequency of each light. We run Algorithm 1 to select the flashing frequencies used by light beacons, assuming the ADC sampling rate of 20 kHz and 128-point FFT. Table 1 lists the frequency of each LED light's light beacons. To support different PWM frequency for each light, we drive the LED lights using five independent 37 V DC power adapters.

Photodiodes. We select SD3410-001 photodiodes for two reasons. First, with the FoV of 90°, all photodiodes in the 3 m × 3 m area can sense light rays from all LED lights that are 2.65 m in height. Second, with the rise time of 75 μs, they support the maximal flashing frequency (3.4 kHz) of the LED lights. We cascade a photodiode and a resistor (10 KΩ) on a mini-breadboard (1.4" × 0.4" × 1.8" in size). Figure 10(b) shows the circuit design. The photodiodes are placed with a uniform interval of 16.7 cm. The resulting photodiode density is sufficient for the posture inference algorithm to track common gestures (e.g., hugging, pointing) with a good accuracy. Capturing finer-grained movements (e.g., finger movements) requires denser photodiodes. We will discuss it in § 7.

Micro-controllers. Each Arduino DUE micro-controller is connected to 12 photodiodes (mini-breadboards). It samples analog voltage numbers of the cascaded resistors on the mini-breadboards (Figure 10(b)) and maps the voltage numbers to integers within 0 and 1023. The Arduino DUE supports 300 kHz ADC sampling rate, sufficient for supporting 20 kHz sampling rate per photodiode. We implement Split-radix Real FFT algorithm [13] on the Arduino board, which processes the Arduino readings for each photodiode using 128 FFT points and computes the frequency power of the PWM frequencies used by the LED lights (Table 1). We select Split-radix Real FFT algorithm because of its efficiency. Running complex FFT subroutine on Arduino boards entails prohibitive overhead – it leads to 2x memory redundancy and wastes half of the processing power on the imaginary part of a complex number, which our input data does not contain.

The Arduino DUE board computes five frequency power numbers (Table 1) for each photodiode, aggregates these numbers for 12 connected photodiodes, and sends them to the server. Each frequency power number is rescaled below 128 and represented by a byte. Thus an Arduino DUE sends 60 bytes to the server each time. We connect the 27 Arduino DUE boards to the server using extended USB cables via series 232 ports (115.2 Kbps data rate).

Server. To aggregate data from all Arduino boards, the server considers the first data packet from each Arduino board within a time window (16.7 ms). We implement the blockage detection mechanism (§ 3.2) and posture inference algorithm (§ 4) in C++ on the server. These two algorithms are run in two separate threads in parallel so that a stream of shadow maps are produced continuously while the skeleton reconstruction thread infers the user posture for each set of concurrent (five) shadow maps. A separate visualization thread then displays the reconstruction results in real time by updating an animated user skeleton (Figure 11).

6. LiSense EXPERIMENTS

We conduct experiments using LiSense testbed, focusing on its skeleton reconstruction accuracy and latency for a wide range of gestures. We also seek to understand the inferred shadow maps accuracy and the impact of practical factors including ambient light, user diversity and clothing, as well as LiSense's energy overhead.

Experimental Setup. We initialize LiSense system in two steps. *First*, we measure the frequency power corresponding to the PWM frequency of each LED L_i (Table 1) at each photodiode p_j when no user stands in the testbed, i.e., P_{ij}^{nonBlock} in Eq. (2). *Second*, we calibrate system parameters by instrumenting the user to stand in the middle of the testbed with arms drooping naturally. When the experiment starts, the user performs a free-form gesture for 2–10 seconds. The server continuously fetches data from all Arduino DUE boards, generates shadow maps, infers user posture for each shadow map, records the 3D vector for each of the five key joints on a user body (backbone joint, left shoulder joint, right should joint, left elbow joint, and right elbow joint), and displays the results as an animated user skeleton. As shadow maps come on the fly, the system continuously tracks user posture overtime and is able to generate 60 inference results per second. By default, all experiments are conducted at night in a typical indoor environment.

To obtain the ground truth of user postures, we set up three high-end SLR cameras (Cannon 60D) on tripods (1.2 m in height), located in the front and on the left and right sides of the user. While a user is performing a gesture, the three cameras capture video at 30 FPS from three perspectives. We then manually label the five key joints every five video frames[5] and calculate the 3D vector of each key point by combining concurrent video frames taken by three cameras. To identify concurrent video frames across cameras, we play a background music during the experiment and later seek the identical audio signal pattern across the video frames from different cameras. We cover a photodiode in the testbed corner before the experiment and remove the cover once each experiment begins. We mark the moment when the corner photodiode becomes unblocked as the starting point for both the video clips and testbed inference results. Human labeling has been the standard technique to seek ground truth in computer vision [26, 41]. In the future, we plan to use VICON to gain finer-grained accuracy.

6.1 Skeleton Reconstruction Accuracy

We first examine LiSense's 3D skeleton reconstruction accuracy with a single user (1.73 m body height, 65 kg body weight, right-handed) performing 20 free-form gestures[6]. These gestures span from single-hand movements (e.g., pushing, pointing, circling, waving), to two-hand movements (e.g., driving, hugging, boxing), to more sophisticated movements involving both user's upper torso and hands (e.g., Combo1: raising and flipping arms, Combo2: pointing and then circling). To examine the skeleton reconstruction accuracy, we represent each key joint as a 3D vector and calculate the

[5]To minimize human labeling errors, we attach flashy, bright strips to the user's key joints during experiments so that they are easily recognizable in video frames.

[6]Current test gestures involve only upper-body movements to avoid accidentally damaging photodiodes on the floor. See more in § 7.

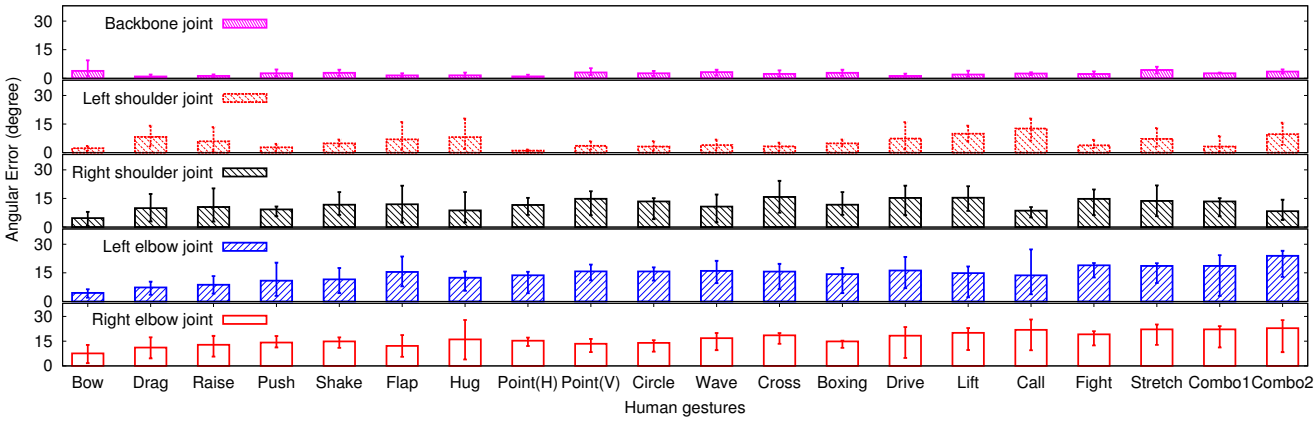


Figure 12: LiSense's skeleton reconstruction accuracy of five key joints on a user body while the user is performing one of the 20 gestures.

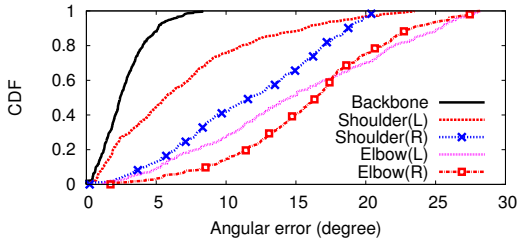


Figure 13: LiSense's skeleton reconstruction accuracy of five key joints on a user body, by aggregating 9292 posture inference results over continuous stream of shadow maps generated by 20 user gestures.

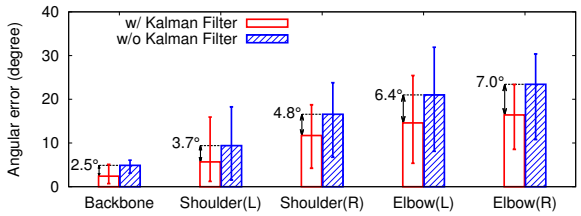


Figure 14: The contribution of the Kalman filter on improving LiSense's reconstruction accuracy. The error bars cover the 90% confidence interval.

absolute angular difference between the inferred and actual vectors for each key joint, referred to as the *3D angular error* of a key joint. This metric has also been widely used by prior work on human motion tracking and skeleton reconstruction [2, 9, 31, 43].

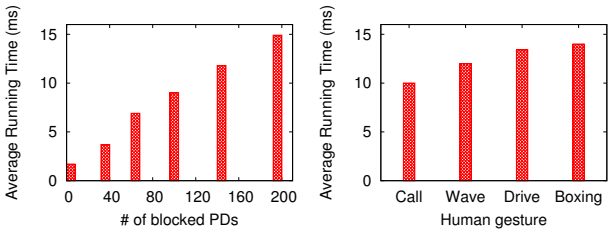
Overall Accuracy. In Figure 12, we plot the mean angular error of each key joint under each test gesture. We also include the error bars covering the 90% confidence interval. Figure 13 further compares the angular errors of the key body joints. Overall, for all gestures, LiSense achieves 10° mean angular error, similar to that of prior gesture tracking systems using RF signals [2, 3]. However, these systems only track a limited set of gestures (e.g., pointing) with a single body part (e.g., arm), while LiSense is capable of reconstructing arbitrary skeleton postures. We recognize that existing vision-based skeleton tracking systems such as Kinect (6.8° – 13.2° angular errors reported in [15]) outperform LiSense in terms of the tracking accuracy. Built upon cameras and depth sensors, these systems are dedicated to skeleton reconstruction and tracking, while LiSense integrates communication and sensing. We will discuss our plan to improve LiSense's accuracy in § 7.

In particular, we observe three key factors that affect LiSense's skeleton reconstruction accuracy under a given photodiode density: *1) Body part size:* LiSense better tracks larger body parts (e.g., backbone joint that corresponds to the user's main body). Because they generate larger shadow than other body parts, their movements lead to more significant differences in shadow maps, allowing our skeleton reconstruction algorithm to infer the 3D vectors more accurately (2° mean angular error). In contrast, reconstructing smaller body parts (e.g., forearms) is more challenging given the limited photodiode density (16° mean angular error). For gestures such as calling, our collected shadow maps do not reflect the shadow change resulted from the forearm movement, because the width of forearm shadow is smaller than the interval between adjacent photodiodes. Therefore the resulting angular errors of shoulder/elbow joints are slightly higher.

2) Movement magnitude: Gestures involving movements at a larger magnitude have smaller angular errors of all key joints, because they lead to more significant change in the shadow maps and are less vulnerable to the occlusion problem. Example gestures include bowing, raising arms, pointing, and dragging. This finding is also exemplified by the results of single-hand gestures (e.g., pushing, pointing, circling). Certain gestures (e.g., stretching) with large movement magnitude have higher angular errors because the shadow exceeds the testbed boundary.

3) Movement speed: High-speed movement can lead to higher angular errors, because LiSense examines the change of the frequency power perceived by each photodiode in a modulation window (6.4 ms). High-speed movement can result into a mixture of blocking and non-blocking states for a single photodiode within a modulation window, which leads to errors in inferred shadow maps and thus larger angular errors. For the same reason, the angular errors of left-side joints are smaller than the right joints for some two-hand gestures (e.g., boxing, fighting), since the right-handed user moves the left hand slightly more slowly. To track faster movements, we can improve the ADC sampling rate and reduce the modulation window size.

Contribution of the Kalman Filter. We further examine the contribution of the Kalman filter (§ 4.2) on reducing the reconstruction error. We analyze all the shadow maps in the experiments and calculate the reconstruction angular error offline without using the Kalman filter for fine-tuning. Figure 14 shows that the fine-tuning reduces the angular error by up to 7 degrees, especially for body parts (e.g., right elbow) with relatively faster movements. The reason is that faster movements lead to more drastic change in the adjacent shadow maps in the time domain. As a result, the raw reconstruction result of LiSense is more vulnerable to bursty errors.



(a) Shadow size (b) Movement speed

Figure 15: The impact of shadow size and movement speed on the running time of user posture inference.

Table 2: Processing time of generating shadow maps of all LEDs. The blockage detection algorithm (§ 3.2) runs instantaneously on the server and thus is not included.

Step	ADC sampling	FFT	Transmission	OS scheduling
Time (ms)	6.4	1.2	4.2	0.2 – 3.2

The Kalman filter smooths out the movement trajectory by modeling the movement as a stochastic process. It filters out the bursty errors and thus improves the reconstruction accuracy.

6.2 Skeleton Reconstruction Latency

We next examine LiSense's skeleton reconstruction latency, consisting of the duration of acquiring shadow maps of all LEDs and the processing time of inferring the user posture based on a single set of shadow maps. Note that these two steps run in parallel in our implementation (§ 5), thus the skeleton reconstruction latency perceived by the end user only depends on the slower step.

Acquiring Shadow Maps. Acquiring shadow maps of all LEDs has four steps: 1) each Arduino DUE samples photodiode data; 2) the Arduino DUE applies FFT over data collected from each photodiode within a modulation window and computes the frequency power of each LED's PWM frequency; 3) the Arduino sends the frequency power numbers of 12 photodiodes to the server; 4) the server runs the blockage detection scheme, aggregates the detection results of all photodiodes, and generates the five shadow maps.

Table 2 lists the latency of each step running at the Arduino DUE board. Overall, generating all five shadow maps takes 16 ms. Specifically, the ADC sampling (step 1) contributes more than half of the delay. Since the ADC sampling rate for each photodiode is 20 kHz, to support 128-point FFT computation in the second step, it takes 6.4 ms for the Arduino DUE to sample all 12 connected photodiodes. The Split-radix Real FFT algorithm is very efficient and takes only 1.2 ms (step 2). The data transmission (step 3) takes 4.2 ms, where the Arduino DUE sends 60 bytes (5 bytes for each photodiode's frequency power numbers) to the server via the series 232 port. Running the blockage detection scheme (step 4) entails negligible delay thanks to the simplicity of the detection scheme (§ 3.2). The only delay in the final step comes from the OS scheduling. It is because the OS (Ubuntu 14.04) of our server does not have a real-time kernel. With 27 threads collecting data from all Arduino boards, the OS can delay some threads when fetching data from a particular Arduino board. We observe that the OS scheduling delay varies from 0.2 – 3.2 ms.

Inferring User Posture. Two factors affect the latency of inferring a user posture based on five shadow maps, which are the shadow size (i.e., the number of photodiodes inside the shadow) and the movement complexity. *First*, to examine the impact of shadow size, we increase the number of blocked photodiodes by increasing the size of the blocking paperboard and plot the average running time of the posture inference in Figure 15 (a). As expected, it takes longer for the inference algorithm to process shadows in larger sizes. However, for all the shadow sizes in our experiments, the posture inference never takes more than 15 ms. LiSense handles large shadow size well because it minimizes the shadow points to process by considering only shadow's boundary points, rather than all its inner points (§ 4). *Second*, we examine four representative gestures with different moving speed and movement complexity, and compare the inference running time. Figure 15 (b) shows that as the gesture becomes more complex with more moving body parts, it takes longer for the skeleton reconstruction algorithm to identify the best-fit skeleton posture. However, the inference delay never exceeds 15 ms even for complex fast-moving gestures (e.g., boxing).

Overall, the processing time of the above two steps shows that LiSense can produce at least 60 user skeleton postures per second on the fly, the same as playing a video at a high frame rate.

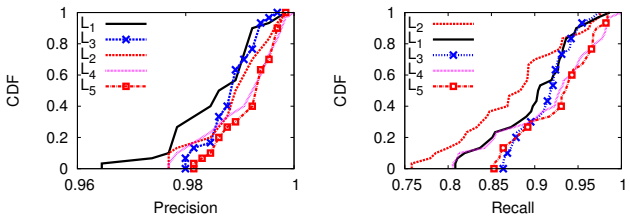


Figure 16: Shadow map accuracy with 30 static user postures.

Table 3: Blockage detector error at a photodiode under varying light incidence angle. The detection accuracy starts to drop only when the incidence angle approaches 45°, half of the photodiode's FoV.

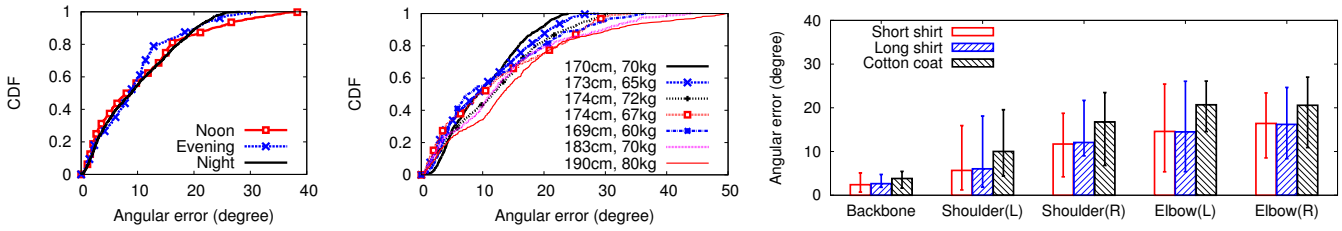
Incidence angle (°)	[0,32)	[32,40.8)	[40.8,43.2)	[43.2, 45)
Blockage detection accuracy (%)	100	95 – 100	90 – 95	58 – 90

6.3 Shadow Map Accuracy

Shadow maps are the key intermediate results for LiSense to reconstruct user's skeleton posture. Now we examine the accuracy of inferred shadow maps and seek to understand the performance of the blockage detection scheme (§ 3.2). We obtain the ground truth shadow maps as follows. We first instruct the user to keep a static posture in the middle of the testbed. We then switch on a single LED light by turns, use an SLR camera (Cannon 60D) mounted at 1.8 m height to take a high-resolution (5184 × 3456) photo of the shadow cast on the floor, and repeat the process for all LED lights. Finally we manually label the shadow maps based on the captured photos and use the labeled shadow maps as the ground truth. We test 30 static postures with a single user.

We compare the inferred shadow maps to the ground truth and evaluate the accuracy using two metrics: 1) *precision*: among all photodiodes, the percentage of photodiodes that are correctly identified as being blocked; and 2) *recall*: among all photodiodes that are actually blocked, the percentage of photodiodes that are correctly identified. We plot the results in Figure 16 for all five LED lights. Overall the shadow maps are fairly accurate: 98% precision and 88-94% recall on average. The shadow maps cast by LED light L_2 have slightly lower recall because L_2 is right above the user (Figure 9(a)) and casts smaller shadow than other lights. Thus its recall results are more sensitive to errors.

As we further analyze the cause of the errors in a shadow map, we find that blockage detection errors occur only when the light rays arrive at the boundary of the photodiode's viewing angle (i.e., the light's incidence angle approaches half of the photodiode's FoV),



(a) Ambient light (b) User diversity (c) User clothing

Figure 17: Impact of practical factors on LiSense's skeleton reconstruction accuracy.

as shown in Table 3. This is because as the light's incidence angle approaches the photodiode's FoV, the photodiode is more sensitive to the hardware noise and its response sensitivity drops drastically. As a result, it barely detects the light intensity changes, leading to the blockage detection errors. We expect that shadow map errors will be diminished when using photodiodes with larger FoVs.

6.4 Practical Considerations

Finally we examine the impact of ambient light, user diversity, and user clothing on LiSense's skeleton reconstruction accuracy. We also examine the energy overhead of human sensing in LiSense.

Ambient Light. We conduct experiments in LiSense at different times of a day, where the intensity of natural light varies from 375 lux at noon, 47 lux at evening, to 0 lux at night. We test a single user performing 10 gestures at each time of a day, and plot the angular errors in Figure 17(a). Overall LiSense's performance is stable in all ambient light conditions. The results obtained at noon have a slightly longer tail, because the higher level of ambient light intensity pushes the photodiodes further up into the saturation region (Figure 3(a)). Thus blocking an LED leads to smaller amplitude change in the perceived light signal, resulting into smaller frequency power changes and slightly higher error in inferred shadow maps and skeleton reconstruction results.

User Diversity. We test LiSense with seven users in different body sizes and shapes. The body height varies from 1.69 m to 1.9 m and the body weight ranges from 60 kg to 80 kg. Each user performs 10 gestures at night and Figure 17(b) compares the angular errors across users. The mean angular error is similar (9° – 16°) across all users, demonstrating LiSense's robustness under diverse users. The taller users have larger tail errors because these users' shadows often exceed the testbed boundary (3 m $\times$ 3 m), especially for gestures with large movement magnitude (e.g., stretching). The incomplete shadows lead to slightly higher errors. We expect these tail errors to disappear once we expand the testbed.

User Clothing. User clothing can also affect LiSense's performance. Consider a user wearing loose clothes, casting a shadow slightly larger than that of the actual user body. This mismatch can confuse LiSense's posture inference algorithm, which is initialized based on the user's body parameters. To examine the impact of user clothing, we repeat the experiments with a single user wearing different types of clothes. We plot the average angular error for each key body joint and the 90% confidence interval in Figure 17(c). We see that bulky or loose clothes (e.g., coat) can decrease the reconstruction accuracy by 6 degrees. In comparison, thin clothes such as short shirts create less interference in the resulting shadow, leading to a higher reconstruction accuracy. To maintain a good reconstruction accuracy under different user clothing, we can adapt user's body parameters to take into account the impact of clothing on the resulting shadow. We plan it for future work.

Energy Overhead. The energy overhead of LiSense comes from acquiring shadow maps from 324 photodiodes and 27 Arduino boards, and running the posture inference algorithm on the server. Our experiments show that acquiring shadow maps consumes 0.4 W atop the 27.4 W idle power of the Arduino boards and photodiodes. The inference algorithm consumes 19 W on our server (DELL M4600 with Ubuntu 14.04) atop the server's idle power. In comparison, the Kinect sensor (Version 2) consumes additional 11 W power for acquiring depth images and the skeleton tracking algorithm consumes 63 W on DELL M4600 with Window 8.1 atop the idle power. Kinect consumes much more power because it needs to capture dense (512 $\times$ 424) depth images and run a heavy classification algorithm on both the CPU and GPU [57].

We can further lower LiSense's energy overhead by reducing its idle power and implementing the inference algorithm on low-power micro-controllers such as Raspberry Pi (2 W). We have successfully migrated LiSense's code to Raspberry Pi. It currently reconstructs the user skeleton at 4 FPS. We plan to further optimize our inference algorithm and reduce the reconstruction latency on Pi.

7. DISCUSSIONS

The initial results from our LiSense testbed are encouraging. But based on our experiences building the LiSense testbed we also recognize several limitations of the existing system and potential applications that motivate future work.

Photodiode Deployment. Deploying hundreds of photodiodes on the floor has been a significant endeavor and learning curve both in terms of the number of man hours to develop such a photodiode floor – connecting 324 photodiodes to 27 micro-controllers – and developing techniques to make our testbed experiments robust and repeatable. We are convinced that the complexity can be eased in the future. Advances in smart fabric [45] will allow us to integrate photodiodes directly into textile. The electrical component in a photodiode is tiny (i.e., 0.5 mm $\times$ 0.5 mm in size) and thus the integration will be feasible in the next few years given the increasing interests in the industry (e.g., Google's Jacquard project [1]). Photodiode-embedded fabric would also allow a much denser deployment of photodiodes – the key to realize reconstructing finer-grained gestures (e.g., finger movements). We also plan to explore the use of photodiodes in wearable devices and in walls to further pursue the vision.

Environmental Factors. Our current testbed is in an open lab space where the test user is the only blocking object. Clearly, this is a simplification of more realistic environments where furniture and other people would block light. We plan to examine settings with other static blocking objects (e.g., sofa, chairs), which can cast their own shadows or block a user's shadow. In this case, we can extract the human gesture by detecting the moving objects in the shadow maps. In the case of multiple users, their shadows would

likely overlap, making the reconstruction of individual user skeletons much more challenging. With denser LEDs on the ceiling, LiSense can extract more details on the human gesture by examining the shadows cast from different viewing angles. These shadows help the system differentiate multiple users. We plan to study these realistic settings as part of our future work.

Improving LiSense Accuracy. LiSense's current skeleton reconstruction accuracy is still below that of vision-based skeleton tracking systems (e.g., Kinect), which use both cameras and depth sensors. However, LiSense holds potential to achieve a greater accuracy. First of all, we can leverage learning algorithms such as the decision forest [57] to segment a shadow map into body parts and optimize each body part individually. The segmentation provides additional information on how and whether a body part moves in the 3D space. It refines the inference and helps the algorithm converge more quickly. Furthermore, we plan to examine placing photodiodes on other planes (e.g., walls) in addition to the floor. Combining the light intensity data from multiple planes allows us to obtain shadow information in multiple dimensions and thus boost the skeleton reconstruction accuracy.

Testbed Maintenance. All photodiodes and control circuits are currently exposed in the air. We observe that practical issue can complicate the maintenance of the testbed; for example, dirt and dust harm electrical component – 41 photodiodes have been damaged in 40 days of testing. Our initial approach has limited gestures to upper-body movements to avoid unexpected damage to photodiodes caused by walking over the fairly fragile photodiode flooring. In the near future, we plan to add a cover made of thin, durable plastic glass (e.g., polycarbonate plastic) over the photodiodes floor so users can stand on the "glass floor" allowing us to experiment with a larger, more realistic set of leg movements – further advancing the gestures we can infer with LiSense.

Potential Applications. With the ability of reconstructing the user skeleton in real time at a high frame rate, LiSense enables new interaction designs and facilitates existing applications. Here we discuss three examples. *First*, LiSense allows users to freely control smart devices (e.g., nano-copters, robot vacuum) in the environment at a fine granularity, using only the ubiquitous light, without any cameras or on-body sensors. *Second*, LiSense facilitates the augmented reality (AR) applications by allowing the users to naturally interact with the virtual scene. AR applications can leverage the reconstructed user skeleton posture to adapt its virtual scenes. *Third*, LiSense serves as the basis for building a passive human behavioral and health monitoring system. By aggregating the reconstructed human posture data over time, we can infer higher-level human behavioral patterns (e.g., gait, movement characteristics) and correlate these patterns to the user's health status.

8. RELATED WORK

VLC Physical Layer. The research group led by Nakagawa at Japan pioneered the early work on VLC [29, 30]. They envisioned the integration of VLC with the power-line communication (PLC) to provide ubiquitous Internet access indoors [29], and first studied VLC propagation properties in indoor settings [30]. Since then, active efforts focus on boosting VLC data rate, by either filtering out the yellow light [33], or designing sophisticated modulation schemes (e.g., CSK [49], OFDM [14]), or enabling parallel data streams via MIMO [68]. The fastest VLC system to date has achieved 10Gbps data rate using arrays of specialized micro-LEDs [59, 60]. Advancements on VLC physical layer techniques are encouraging and LiSense can be built atop them by seamlessly integrating light beacons with VLC communication.

VLC Applications. Current VLC applications span from indoor communication [38], indoor localization [28, 32, 36, 69], screen-camera communication [7, 18, 22, 23, 37, 44, 61], vehicular networks [6, 39], LED-to-LED communication [12, 54], underwater communication [53], to in-flight entertainment [47]. Our work broadens VLC use cases and explores the feasibility of using VLC for fine-grained, real-time human sensing, enabling a new paradigm of user interaction. Existing work on VLC indoor localization [28, 32, 36, 69] is similar in the spirit by sensing user location. LiSense differs in that it enables much finer-grained user sensing and tracks a wide range of user's gestures.

Gesture Recognition and Reconstruction. We divide existing work into two categories. The first type of work uses sound waves [17] or RF signals [2, 3, 4, 46, 62, 63] to recognize and track gestures. These systems focus on either classifying a limited set of gestures or activities [4, 17, 46, 63], or tracking the movement of a single body part [2, 3], or requiring on-body sensors [62]. LiSense differs in that it reconstructs the user's skeleton posture in the 3D space in real time. Thus it is capable of tracking any gestures, without requiring the user to wear on-body sensors.

The second type of work [11, 20, 21, 24, 41] uses cameras and/or depth sensors to capture high-resolution video frames and build dedicated sensing systems. They apply computer vision algorithms to track or reconstruct gestures. Overall, these methods typically involve a heavy computation, which limits their achieved frame rates (30 FPS for Kinect). Furthermore, these methods often require an exhaustive training to achieve a reasonable accuracy. The skeleton tracking algorithm [57] used in Kinect relies on hundreds of thousands of training images to build the classifier. Finally, capturing and storing the raw camera data raises privacy concerns since these sensitive data can be leaked to the adversary [52, 64]. In comparison, LiSense integrates sensing and communication. It uses low-resolution shadow maps to reconstruct the user skeleton. The inference algorithm is lightweight and achieves 60 FPS frame rate in our current implementation.

9. CONCLUSION

In this paper, we discussed the design, deployment, and evaluation of LiSense, the first human gesture tracking system purely powered by visible light communication (VLC) – light is used for both human sensing and communication. LiSense uses light beacons to recover the shadow pattern cast by individual light sources. It infers 3D human postures at 60 Hz in real time with mean 10° angular error in 3D skeleton reconstruction. LiSense lies in the intersection of wireless networking, computer vision, and HCI. It enables new VLC applications and new forms of interactions not possible before.

10. ACKNOWLEDGMENTS

We sincerely thank anonymous reviewers for their valuable comments that helped improve the paper. We also thank Guanyang Yu and Xiaole An for their contribution on visualizing the user skeleton, Xing-Dong Yang for the support on Kinect experiments, Will Campbell and DartNets Lab members Fanglin Chen, Peilin Hao, and Rui Wang for their help on our experiments. This work is supported in part by the National Science Foundation under grant CNS-1421528. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the funding agencies or others.

11. REFERENCES

[1] Google Project Jacquard. https://www.google.com/atap/project-jacquard/.

[2] ADIB, F., KABELAC, Z., AND KATABI, D. Multi-Person Motion Tracking via RF Body Reflections. In *Proc. of NSDI* (2015).

[3] ADIB, F., KABELAC, Z., KATABI, D., AND MILLER, R. C. 3D Tracking via Body Radio Reflections. In *Proc. of NSDI* (2014).

[4] ADIB, F., AND KATABI, D. See through walls with WiFi! In *Proc. of SIGCOMM* (2013).

[5] AFGANI, M. Z., HAAS, H., ELGALA, H., AND KNIPP, D. Visible light communication using OFDM. In *Proc. of TRIDENTCOM* (2006).

[6] ARAI, S., ET AL. Experimental on hierarchical transmission scheme for visible light communication using LED traffic light and high-speed camera. In *Proc. of VTC* (2007).

[7] ASHOK, A., ET AL. Challenge: Mobile optical networks through visual MIMO. In *Proc. of MobiCom* (2010).

[8] ASSOCIATION, I. S., ET AL. IEEE Std. for Local and metropolitan area networks-Part 15.7: Short-Rang Wireless Optical Communication Using Visible Light. *IEEE Computer Society* (2011).

[9] BEAUDOIN, P., POULIN, P., AND VAN DE PANNE, M. Adapting wavelet compression to human motion capture clips. In *Proc. of GI* (2007).

[10] CHEN, V. C., LI, F., HO, S.-S., AND WECHSLER, H. Analysis of micro-doppler signatures. *IEE Proceedings-Radar, Sonar and Navigation 150*, 4 (2003), 271–276.

[11] CHEUNG, G. K., KANADE, T., BOUGUET, J.-Y., AND HOLLER, M. A real time system for robust 3D voxel reconstruction of human motions. In *Proc. of CVPR* (2000).

[12] DIETZ, P., YERAZUNIS, W., AND LEIGH, D. Very low-cost sensing and communication using bidirectional LEDs. In *Proc. of UbiComp* (2003).

[13] DUHAMEL, P., AND HOLLMANN, H. Split radix FFT algorithm. *Electronics letters 20*, 1 (1984), 14–16.

[14] ELGALA, H., MESLEH, R., HAAS, H., AND PRICOPE, B. OFDM visible light wireless communication based on white LEDs. In *Proc. of VTC* (2007).

[15] FERNANDEZ-BAENA, A., SUSÍN, A., AND LLIGADAS, X. Biomechanical validation of upper-body and lower-body joint movements of kinect motion capture data for rehabilitation treatments. In *Proc. of INCoS* (2012).

[16] GHADDAR, M., TALBI, L., AND DENIDNI, T. Human body modelling for prediction of effect of people on indoor propagation channel. *Electronics Letters 40*, 25 (2004), 1592–1594.

[17] GUPTA, S., MORRIS, D., PATEL, S., AND TAN, D. SoundWave: Using the Doppler effect to sense gestures. In *Proc. of CHI* (2012).

[18] HAO, T., ZHOU, R., AND XING, G. COBRA: Color barcode streaming for smartphone systems. In *Proc. of MobiSys* (2012).

[19] HENRY, P., KRAININ, M., HERBST, E., REN, X., AND FOX, D. RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments. In *Proc. of ISER* (2010).

[20] HERDA, L., FUA, P., PLANKERS, R., BOULIC, R., AND THALMANN, D. Skeleton-based motion capture for robust reconstruction of human motion. In *Proc. of International Conference on Computer Animation* (2000).

[21] HOWE, N. R., LEVENTON, M. E., AND FREEMAN, W. T. Bayesian Reconstruction of 3D Human Motion from Single-Camera Video. In *Proc. of NIPS* (1999).

[22] HU, W., GU, H., AND PU, Q. LightSync: Unsynchronized visual communication over screen-camera links. In *Proc. of MobiCom* (2013).

[23] HU, W., MAO, J., HUANG, Z., XUE, Y., SHE, J., BIAN, K., AND SHEN, G. Strata: Layered Coding for Scalable Visual Communication. In *Proc. of MobiCom* (2014).

[24] IZADI, S., KIM, D., HILLIGES, O., MOLYNEAUX, D., NEWCOMBE, R., KOHLI, P., SHOTTON, J., HODGES, S., FREEMAN, D., DAVISON, A., ET AL. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In *Proc. of UIST* (2011).

[25] JERRI, A. J. The Shannon sampling theorem–Its various extensions and applications: A tutorial review. *Proc. of the IEEE* (1977).

[26] JHUANG, H., GALL, J., ZUFFI, S., SCHMID, C., AND BLACK, M. J. Towards understanding action recognition. In *Proc. of ICCV* (2013).

[27] KIM, D., HILLIGES, O., IZADI, S., BUTLER, A. D., CHEN, J., OIKONOMIDIS, I., AND OLIVIER, P. Digits: Freehand 3D Interactions Anywhere Using a Wrist-worn Gloveless Sensor. In *Proc. of UIST* (2012).

[28] KIM, H.-S., ET AL. An indoor visible light communication positioning system using a RF carrier allocation technique. *Journal of Lightwave Technology 31*, 1 (2013), 134–144.

[29] KOMINE, T., AND NAKAGAWA, M. Integrated system of white LED visible-light communication and power-line communication. *IEEE Transactions on Consumer Electronics 49*, 1 (2003), 71–79.

[30] KOMINE, T., AND NAKAGAWA, M. Fundamental analysis for visible-light communication system using LED lights. *IEEE Transactions on Consumer Electronics 50*, 1 (2004), 100–107.

[31] KROSSHAUG, T., AND BAHR, R. A model-based image-matching technique for three-dimensional reconstruction of human motion from uncalibrated video sequences. *Journal of biomechanics 38*, 4 (2005), 919–929.

[32] KUO, Y.-S., PANNUTO, P., HSIAO, K.-J., AND DUTTA, P. Luxapose: Indoor positioning with mobile phones and visible light. In *Proc. of MobiCom* (2014).

[33] LE-MINH, H., ET AL. 100-Mb/s NRZ visible light communications using a postequalized white LED. *Photonics Technology Letters, IEEE 21*, 15 (2009), 1063–1065.

[34] LE MINH, H., O'BRIEN, D., FAULKNER, G., ZENG, L., LEE, K., JUNG, D., OH, Y., AND WON, E. T. 100-Mb/s NRZ visible light communications using a postequalized white LED. *Photonics Technology Letters, IEEE 21*, 15 (2009), 1063–1065.

[35] LEE, K., AND PARK, H. Modulations for visible light communications with dimming control. *Photonics Technology Letters, IEEE 23*, 16 (2011), 1136–1138.

[36] LI, L., HU, P., PENG, C., SHEN, G., AND ZHAO, F. Epsilon: A visible light based positioning system. In *Proc. of NSDI* (2014).

[37] LI, T., AN, C., XIAO, X., CAMPBELL, A. T., AND ZHOU, X. Real-Time Screen-Camera Communication Behind Any Scene. In *Proc. of MobiSys* (2015).

[38] LITTLE, T. D. C., ET AL. Using LED lighting for ubiquitous indoor wireless networking. In *Proc. of WiMob* (2008).

[39] LIU, C. B., SADEGHI, B., AND KNIGHTLY, E. W. Enabling vehicular visible light communication (V2LC) networks. In *Proc. of VANET* (2011).

[40] MCKINNEY, J. C., AND HOPKINS, C. D. R. ATSC digital television standard. *Advanced Television System Committee* (1995).

[41] MOESLUND, T. B., HILTON, A., AND KRÜGER, V. A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding 104*, 2 (2006), 90–126.

[42] MOHR, R., QUAN, L., AND VEILLON, F. Relative 3D reconstruction using multiple uncalibrated images. *The International Journal of Robotics Research 14*, 6 (1995), 619–632.

[43] MOSCHINI, D., AND FUSIELLO, A. Tracking human motion with multiple cameras using an articulated model. In *Computer Vision/Computer Graphics Collaboration Techniques*. Springer, 2009, pp. 1–12.

[44] PERLI, S. D., AHMED, N., AND KATABI, D. PixNet: Interference-free wireless links using LCD-camera pairs. In *Proc. of MobiCom* (2010).

[45] POST, E. R., ORTH, M., RUSSO, P. R., AND GERSHENFELD, N. E-broidery: Design and fabrication of textile-based computing. *IBM Systems Journal 39*, 3.4 (2000), 840–860.

[46] PU, Q., GUPTA, S., GOLLAKOTA, S., AND PATEL, S. Whole-home gesture recognition using wireless signals. In *Proc. of MobiCom* (2013).

[47] QUINTANA, C., ET AL. Reading lamp-based visible light communication system for in-flight entertainment. *Consumer Electronics, IEEE Transactions on 59*, 1 (2013), 31–37.

[48] RAJAGOPAL, N., LAZIK, P., AND ROWE, A. Visual light landmarks for mobile devices. In *Proc. of IPSN* (2014).

[49] RAJAGOPAL, S., ROBERTS, R., AND LIM, S.-K. IEEE 802.15.7 visible light communication: modulation schemes and dimming support. *Communications Magazine, IEEE 50*, 3 (2012), 72–82.

[50] RYCKAERT, J., DE DONCKER, P., MEYS, R., DE LE HOYE, A., AND DONNAY, S. Channel model for wireless communication around human body. *Electronics Letters 40*, 9 (2004), 543–544.

[51] SAXENA, A., CHUNG, S. H., AND NG, A. Y. 3-D depth reconstruction from a single still image. *International journal of computer vision 76*, 1 (2008), 53–69.

[52] SBÎRLEA, D., BURKE, M. G., GUARNIERI, S., PISTOIA, M., AND SARKAR, V. Automatic detection of inter-application permission leaks in android applications. *IBM Journal of Research and Development 57*, 6 (2013), 10–1.

[53] SCHILL, F., ZIMMER, U. R., AND TRUMPF, J. Visible spectrum optical communication and distance sensing for underwater applications. In *In Proc. of Australasian Conference on Robotics and Automation* (2004).

[54] SCHMID, S., CORBELLINI, G., MANGOLD, S., AND GROSS, T. R. LED-to-LED visible light communication networks. In *Proc. of MobiHoc* (2013).

[55] SEGEN, J., AND KUMAR, S. Shadow gestures: 3D hand pose estimation using a single camera. In *Proc. of CVPR* (1999).

[56] SHOEMAKER, G., TANG, A., AND BOOTH, K. S. Shadow reaching: a new perspective on interaction for large displays. In *Proc. of UIST* (2007).

[57] SHOTTON, J., ET AL. Real-time human pose recognition in parts from single depth images. In *Proc. of CVPR* (2011).

[58] SILAGHI, M.-C., PLÄNKERS, R., BOULIC, R., FUA, P., AND THALMANN, D. Local and global skeleton fitting techniques for optical motion capture. In *Modelling and Motion Capture Techniques for Virtual Environments*. Springer, 1998, pp. 26–40.

[59] TSONEV, D., ET AL. A 3-Gb/s Single-LED OFDM-based Wireless VLC Link Using a Gallium Nitride μLED. *Photonics Technology Letters, IEEE PP*, 99 (2014), 1–1.

[60] WALL, M. 'Li-fi' via LED light bulb data speed breakthrough. BBC News, 2013.

[61] WANG, A., MA, S., HU, C., HUAI, J., PENG, C., AND SHEN, G. Enhancing Reliability to Boost the Throughput over Screen-camera Links. In *Proc. of MobiCom* (2014).

[62] WANG, J., VASISHT, D., AND KATABI, D. RF-IDraw: Virtual Touch Screen in the Air Using RF Signals. In *Proc. of SIGCOMM* (2014).

[63] WANG, Y., LIU, J., CHEN, Y., GRUTESER, M., YANG, J., AND LIU, H. E-eyes: Device-free Location-oriented Activity Identification Using Fine-grained WiFi Signatures. In *Proc. of MobiCom* (2014).

[64] WEINBERG, Z., CHEN, E. Y., JAYARAMAN, P. R., AND JACKSON, C. I still know what you visited last summer: Leaking browsing history via user interaction and side channel attacks. In *Proc. of IEEE Symposium on Security and Privacy* (2011).

[65] WELCH, T., MUSSELMAN, R., EMESSIENE, B., GIFT, P., CHOUDHURY, D., CASSADINE, D., AND YANO, S. The effects of the human body on UWB signal propagation in an indoor environment. *IEEE Journal on Selected Areas in Communications 20*, 9 (2002), 1778–1782.

[66] WHITAKER, R. T. A level-set approach to 3D reconstruction from range data. *International journal of computer vision 29*, 3 (1998), 203–231.

[67] YEUNG, K.-Y., KWOK, T.-H., AND WANG, C. C. Improved Skeleton Tracking by Duplex Kinects: A Practical Approach for Real-Time Applications. *Journal of Computing and Information Science in Engineering 13*, 4 (2013), 041007.

[68] ZENG, L., ET AL. High data rate multiple input multiple output (MIMO) optical wireless communications using white LED lighting. *IEEE Journal on Selected Areas in Communications 27*, 9 (2009), 1654–1662.

[69] ZHANG, W., AND KAVEHRAD, M. Comparison of VLC-based indoor positioning techniques. In *Proc. of SPIE* (2013).

[70] ZHANG, Z., ET AL. I Am the Antenna: Accurate Outdoor AP Location Using Smartphones. In *Proc. of MobiCom* (2011).

[71] ZHOU, X., AND CAMPBELL, A. Visible Light Networking and Sensing. In *Proc. of HotWireless* (2014).