Practical Human Sensing in the Light

Tianxing Li, Qiang Liu, and Xia Zhou
Department of Computer Science, Dartmouth College, Hanover, NH
{tianxing, qliu, xia}@cs.dartmouth.edu

ABSTRACT

We present StarLight, an infrastructure-based sensing system that reuses light emitted from ceiling LED panels to reconstruct fine-grained user skeleton postures continuously in real time. It relies on only a few (e.g., 20) photodiodes placed at optimized locations to passively capture low-level visual clues (light blockage information), with neither cameras capturing sensitive images, nor on-body devices, nor electromagnetic interference. It then aggregates the blockage information of a large number of light rays from LED panels and identifies best-fit 3D skeleton postures. StarLight greatly advances the prior light-based sensing design by dramatically reducing the number of intrusive sensors, overcoming furniture blockage, and supporting user mobility. We build and deploy StarLight in a 3.6 m × 4.8 m office room, with customized 20 LED panels and 20 photodiodes. Experiments show that StarLight achieves 13.6° mean angular error for five body joints and reconstructs a mobile skeleton at a high frame rate (40 FPS). StarLight enables a new unobtrusive sensing paradigm to augment today's mobile sensing for continuous and accurate behavioral monitoring.

Keywords

Visible light communication; skeleton tracking; localization; sensing

1. INTRODUCTION

Accurate, continuous behavioral sensing is crucial to our health and daily life. It helps detect early symptoms of health issues and disorders [17, 38] and foster behavioral changes to cultivate healthy lifestyles [12, 14, 29, 34, 38, 45, 49]. The technology shift is arriving with the advent of wearable and wellness sensors (e.g., Fitbit, Apple Watch). These sensing devices monitor our activities and health status at an unprecedented level of details (e.g., footsteps, heart beats).

However, these mobile sensing devices alone are still fundamentally limited. Not only do we have to constantly wear or carry and regularly charge them, but also they fall far short on gaining an accurate and complete view of our behaviors. Devices like Fitbit and Apple Watch can only infer physical movement using crude accelerometers. Similarly, not only are heartbeat sensors error-prone (especially during movement), but also they say nothing about the causes that trigger an increase in heart rate, which can be due to exercising, or potential health issues (e.g., stress) that need immediate intervention. This lack of the semantic context prevents these wearable sensors from bridging the gap between local measurements and a holistic view of our behaviors.

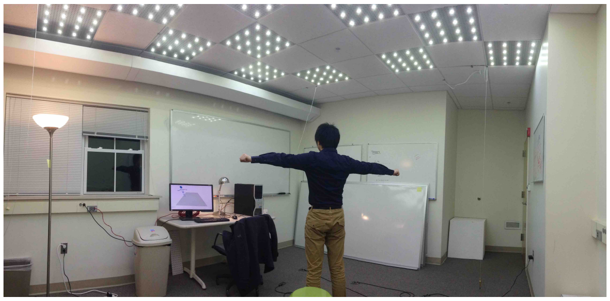


Figure 1: StarLight reuses ceiling LED panels together with a few photodiodes to reconstruct a mobile user's skeleton in a normal office setting with furniture.

To bring human sensing to the next level, we have to augment per-user, local sensors with environmental sensing data. Current methods all have significant drawbacks. They either require capturing raw images and video [11, 19, 21, 23, 42, 54] that present severe privacy risks involving the leaking of sensitive images [1] [52, 66], or reuse ambient radio frequency (RF) signals that are vulnerable to electromagnetic interference and offer only coarse sensing granularity (e.g., classifying pre-defined gestures and activities, tracking a single body part) [6, 7, 8, 18, 48, 58, 63, 65].

In this paper, we pursue the sensing paradigm that exploits the ubiquitous light as a passive sensing medium that accurately senses what we do at the infrastructure level, with neither on-body sensors, nor cameras capturing sensitive images/videos, nor electromagnetic interference. A light-based sensing system consists of Light-Emitting-Diodes (LEDs) on the ceiling and low-cost (<$3) light sensors (photodiodes, 3 mm in radius) in the environment (e.g., embedded in the carpet using smart textile [47]). These light sensors passively collect the shadow information created by our body blocking the lights and recover our behaviors continuously. By collecting such low-level visual clues (see examples in Figure 7(b)) and removing the need of high-fidelity sensors such as cameras, light-based sensing much better protects user privacy. It

MobiSys'16, June 25-30, 2016, Singapore, Singapore
© 2016 ACM. ISBN 978-1-4503-4269-8/16/06...$15.00
DOI: http://dx.doi.org/10.1145/2906388.2906401

[1] Although one can store only features of interests extracted from camera images to alleviate the privacy concern, such process is regularly conducted in software and thus still allows potential attacks for the adversary.

also entails a low cost – it reuses the ubiquitous lighting infrastructure without the need to replace LEDs but only connecting LEDs to low-cost micro-controllers (e.g., FPGAs, Arduino boards).

Initial efforts [36] have demonstrated the feasibility of light-based sensing, however, several key limitations remain. *First*, prior design needs hundreds of photodiodes to acquire shadow information even for a 3 m × 3 m area. The overhead of photodiode deployment prevents its immediate adoption. *Second*, it assumes an open space with the user as the only light-blocking object, whereas in practice, other objects (e.g., furniture) in the environment can either block the shadow or create overlapping shadows, making it much more challenging to recover user behaviors. *Finally*, prior system works only for a single static user with known location and orientation, dictating rather limited working scenarios without user mobility.

The focus of this paper is to overcome the above limitations and further unleash the practical potential of light-based sensing in typical indoor settings. To this end, we present *StarLight, the first light-based sensing system that uses only a few (e.g., 20) photodiodes to reconstruct moving human skeletons in real time, even in the presence of furniture and other blocking objects*. The underlying principle of StarLight is a new sensing architecture that reuses ubiquitous LED panels on the ceiling. Designed to generate homogeneous and natural light, each LED panel contains arrays of LED chips inside.[2] Each LED chip functions as a point light source. These LEDs together provide a large number of light paths for analysis. Since the blockage of each light path is independent of which end of the path is LED and which end is the photodiode, by separating light rays from different LED chips, each photodiode can recover a *virtual shadow map*, which would have been projected on the ceiling if this photodiode were an LED and LED chips on the ceiling were photodiodes. By sprinkling a small number of photodiodes in the environment, we can collect virtual shadow maps from these photodiodes' viewing points. These virtual shadow maps allow the system to reconstruct a user's moving skeleton. Compared to the prior design [36] with a few LEDs and many photodiodes, this new sensing architecture drastically reduces the number of photodiodes to deploy, making the sensing system much easier to be integrated into existing and future lighting infrastructure.[3]

The new sensing architecture, on the other hand, also brings unique design challenges not present before. *First*, it is non-trivial to separate light rays from a large number of LEDs on the ceiling using low-cost photodiodes and micro-controllers. Prior design [36] requires each LED to flash at a unique frequency and thus cannot directly scale up to dense LEDs. *Second*, given the small number of photodiodes and their limited viewing angles, optimizing their placements is crucial to the sensing performance. Our experiments show that simple placement strategies (e.g., uniform) degrades the sensing accuracy. The placement optimization problem faces an exponential search space and needs to take into account practical factors such as the non-uniform placement of LED panels, furniture blockage, and user mobility. *Third*, the reconstruction algorithm deals with the unknown user location, as well as virtual shadows that can be incomplete due to the LED panel placement and furniture blockage. Additionally, it has to entail low complexity to realize real-time skeleton reconstruction.

StarLight addresses these challenges as follows. *First*, it reuses light beacons over time and thus reduces the number of concurrent light beacon frequencies that need to be separated. It also includes synchronization schemes and judicious light frequency assignment to facilitate light beacon identification at the photodiodes. *Second*, it turns the problem of optimizing photodiode placement into a submodular maximization and designs an efficient greedy solution with $(1 - 1/e)$ approximation ratio. It considers the practical layout of main furniture and LED panels as input and seeks the optimized sensor placement to minimize the impact of furniture blockage on the sensing performance. *Finally*, to support user mobility and orientation change, StarLight tracks user's location and orientation based on coarse-grained body features extracted from binary virtual shadow maps. It aggregates the virtual shadow maps from all photodiode sensors to overcome the incompleteness and the extremely low resolution of individual maps. It also leverages user's movement continuity to reduce the search space and identify the best-fit 3D skeleton posture.

StarLight Testbed. We implement and deploy StarLight in a 3.6 m × 4.8 m office room with normal furniture layout (Figure 8(a)). The testbed consists of 20 customized LED panels (Figure 8(b)) and 20 low-cost photodiodes [4] placed on the floor based on the greedy placement strategy. We design and fabricate the LED panels, each containing 16 off-the-shelf LEDs modulated by an FPGA independently (Figure 10). The photodiodes connect to Arduino Due micro-controllers, which sample light intensity data, separate light rays from different LEDs, and transfer results to a server. The server detects the blockage of each light ray, recovers virtual shadow maps, runs the skeleton reconstruction algorithm, and visualizes the reconstructed skeleton on a monitor in real time. Figure 1 is a snapshot of StarLight in action.

We evaluate StarLight under diverse settings and obtain the following key findings:

- StarLight achieves 13.6° mean angular error for 3D reconstructing five key body joint, comparable to the prior design while addressing several practical concerns (sensor deployment overhead, furniture blockage, user mobility);
- StarLight tracks a mobile skeleton with 2D localization error of 4 cm on average and 9.7 cm at the 95%-percentile;
- StarLight produces a reconstructed 3D skeleton within 25 ms and visualizes the human skeleton at 40 FPS, higher than that of Kinect (30 FPS);
- StarLight maintains a stable performance under different ambient light and furniture layout conditions.

2. LIGHT-BASED SENSING: CONCEPT AND CHALLENGES

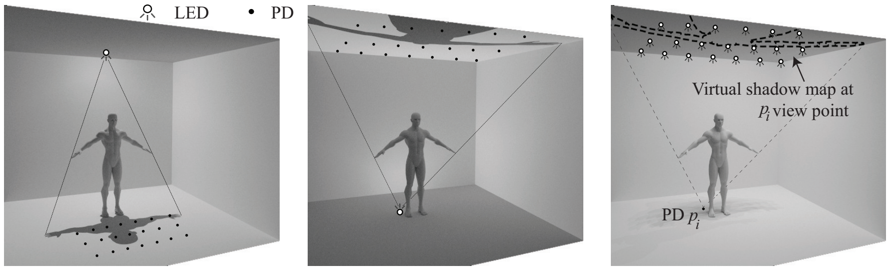
The goal of light-based sensing is to turn everyday lighting into a powerful, accurate sensing medium that can reconstruct our whole-body gestures and infer our detailed activities. In this section, we start with the background of light-based sensing and the limitations of prior design. We then describe a new sensing architecture that helps overcome these limitations, followed by the practical challenges that we face to realize the new architecture.

2.1 Background of Light-Based Sensing

Concept. Light-based sensing is driven by a simple observation: shadows. Because the wavelength of visible light is measured in nanometers, any opaque macroscopic object (e.g., a user body) can block the light and cast a silhouette behind the object. The resulting shadow is essentially a 2D projection of the 3D object. Take the human body as an example, as the body moves and gestures, it blocks different light rays and its shadow shape changes at the light speed. Thus, by continuously collecting and analyzing the stream

[2] As an example, an off-the-shelf direct-lit LED panel [1] contains 132 LED chips in a 61 cm × 61 cm area.

[3] LED panels are envisioned to be the future of indoor lighting because of their efficiency and seamless ceiling integration [3].



(a) Shadow on the floor (b) Shadow on the ceiling (c) Virtual shadow on the ceiling

Figure 2: Comparison of the sensing architecture. Prior design relies on few LEDs and many photodiodes (PDs) to collect shadow information on the floor (a). To minimize the photodiode deployment overhead, the new sensing architecture aims to recover the shadow projected on the ceiling (b), by reusing arrays of LED chips realized as LED panels on the ceiling (c).

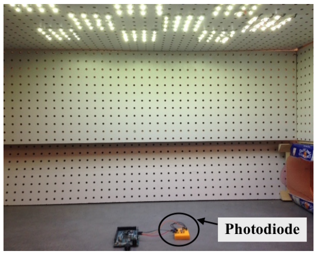


Figure 3: A table-top testbed measured 30 cm × 60 cm × 50 cm in size, with 9 customized LED mini-panels on the ceiling. Each LED mini-panel contains 16 CREE XML LED chips controlled by an FPGA board.

of shadows cast on the floor, we can infer user's 3D postures over time and recover the activities.

Prior design LiSense [36] has demonstrated the feasibility of light-based sensing. As illustrated in Figure 2(a), the system architecture of LiSense consists of dense (e.g., 36 photodiodes per m^2) photodiodes on the floor to capture the shadow information and a small (e.g., 5) number of LED lights on the ceiling. LiSense comprises two systems pieces: shadow acquisition and posture reconstruction. Shadow acquisition disambiguates the diluted and complex shadow patterns under multiple lights and acquires the shadow component corresponding to each light. Empowered by Visible Light Communication (VLC), LiSense leverages light beacons implemented with VLC to separate light rays from different LEDs and ambient light. Each LED emits light beacons by flashing at a unique high frequency imperceptible to human eyes. Each photodiode transforms the perceived light intensity values over time to the frequency domain and detects the blockage of the light path to each LED. Aggregating the blockage results of all photodiodes recovers the shadow maps cast by all LEDs. Based on these shadow maps, posture reconstruction infers a human 3D skeleton best matching the observed shadow maps cast in different directions.

Current Limitations. Although prior design provides promising sensing accuracy, its main practical concern is the deployment of dense photodiodes on the floor. Even if the use of smart textile allows integrating these tiny photodiodes with the environment (e.g., rugs, carpet) in an unobtrusive manner, other objects such as furniture can easily block user's shadows and severely interfere with the acquisition of shadow information. Furthermore, the prior posture reconstruction algorithm assumes a single static user with known location and orientation, unable to support user movement and orientation change.

2.2 Rethinking the Sensing Architecture

The need to greatly drive down the overhead of photodiode deployment motivates us to rethink the architecture of the sensing system. Instead of focusing on the shadow cast on the floor, we turn our attention to the shadow projected on the ceiling if LEDs are deployed on the floor. Consider a single LED on the floor (Figure 2(b)), the ceiling shadow created by the human body blocking this LED reflects how the body blocks each light ray emitting from the floor LED to the ceiling. Since the blockage of a light path is independent of which end of the path is the LED, we can switch the two ends of each light path, i.e., placing the LED as the ceiling end of each light path and the photodiode as the floor end, while still being able to infer the blockage of each light path. This switch leads us to a new sensing architecture with dense LEDs on the ceiling and a small number (e.g., dozens) of photodiodes sprinkled in the environment. Each photodiode separates the light rays from all ceiling LEDs, infers the blockage of each light path, and recovers a *virtual shadow map* that would have been projected on the ceiling if the photodiode were an LED and ceiling LEDs were photodiodes (Figure 2(c)). By aggregating the virtual shadow maps from all photodiodes' viewing points, we can infer the user's 3D posture, similarly to the prior design.

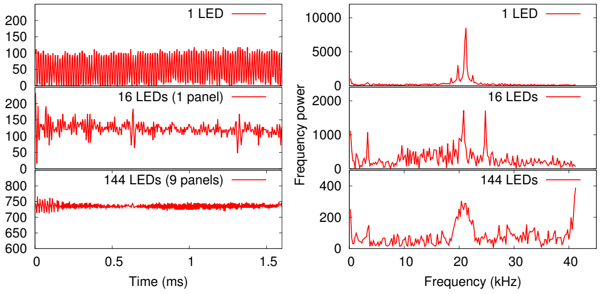
Benefit. This new sensing architecture allows more seamless integration with the existing indoor infrastructure. Leveraging the dense array of LEDs on the ceiling, it significantly reduces the number of photodiodes to deploy on the floor. More importantly, LED arrays can be realized as LED panels on the ceiling, which is envisioned to be the future of indoor lighting because of its efficiency and ability of generating natural light [3]. We can reuse these LED panels without the need to replace the LED chips but only attach to them a modulation unit (e.g., FPGA control board) that controls the input current to each LED.

2.3 Practical Challenges

To build a practical light-based sensing system based on the new sensing architecture, we face three main challenges. They arise from dealing with the *large* number of LED chips on the ceiling, optimizing the placement of the *small* number of photodiodes in the environment, and supporting user mobility while maintaining accurate posture sensing.

Dense LEDs. With multiple LED panels on the ceiling, adjacent LED chips can be only a few centimeters away. Commercial LED typically has a Field-of-View (FoV) of 90° – 120° to maximize the illumination region. Thus, a photodiode on the floor perceives light rays from a large number (e.g., ∼300) of LEDs. These many LEDs present new challenges when it comes to separating light rays. Prior light beacon design (see § 2.1) falls short.

To examine the impact of the number of LEDs, we build a scaled-down table-top testbed resembling a miniaturized room (Figure 3). It is made of wooden frames and plastic plates as walls and the ceiling. We attach 9 LED mini-panels to the ceiling, where each panel is 10 cm × 10 cm in size and contains 16 LED chips (CREE XML). Each panel is connected to an FPGA board (Digilent Basys 3), which modulates the input current to each LED independently. We build other electrical components (e.g., MOSFET, resistor) into a Printed Circuit Board (PCB) (Figure 10(c)) to minimize the electrical noise. We implement the prior light beacon design in the FPGA board. Each LED is assigned with a different light beacon frequency, ranging from 20.8 KHz to 41.5 KHz. We place a photodiode (OPT101 [4]) on the table center and fetch the light signals



(a) Time domain (b) Frequency domain

Figure 4: The impact of the number of LEDs on separating light rays using the prior light beacon design [36].

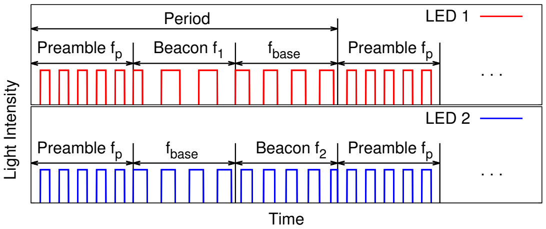


Figure 5: The light beacon design in StarLight. All LEDs first flash at the preamble frequency f_p for synchronization and then enter the beacon slots. Each LED i flashes at its light beacon frequency f_i only in its assigned beacon slot b_i, and at the based frequency f_{base} in other beacon slots. f_i and b_i together uniquely identify LED i.

using an Arduino DUE micro-controller, with the same setup as the previous study [36]. The Arduino reading (an integer within [0, 1023]) reflects the light intensity perceived by the photodiode.

Figure 4 plots the time series of Arduino's readings when we switch on different number of LED chips, as well as the results of transforming these values to the frequency domain after applying Fast Fourier Transform (FFT). Our key observation is that as the number of LEDs increases, the light intensity values perceived by the photodiode exhibit smaller variations, resulting into much lower frequency powers at light beacon frequencies. Therefore, the detection of light beacons is less reliable. Our further analysis shows that the variation decrease is due to the photodiode's response time, which measures the speed of the photodiode to change the output current upon light intensity change. When combining many asynchronous light rays flashing at different frequencies (i.e., different frequency of rising and falling edges), the minimal time interval T_{min} between adjacent rising or falling edges is small. Additionally, the asynchronous flashing of LEDs further shrinks T_{min} in practice.[4] For example, with all LEDs switched on in our table-top testbed, T_{min} is only 46 ns, much shorter than the response time (9 μs) of common low-cost photodiodes. As a result, the photodiode is unable to reach its final output current, leading to a lower Arduino reading and smaller variation. We can mitigate the problem using photodiodes with shorter response time, which, however, requires a smaller sensitivity area inside the photodiode, lowering the sensing gain and distance. As an example, the high-end photodiode [2] we have measured achieves 10-ns response time, whereas its sensing distance is only 30 cm, not applicable in our sensing scenario.

Sparse Photodiodes. Another challenge is to realize accurate gesture reconstruction using only a few photodiodes. The small quantity of photodiodes emphasizes the importance of their placement for the sensing performance. Ideally, photodiodes should be placed to maximally capture the light ray blockage information created by the user body. Their placement, however, is complicated by the following practical factors. *First*, other blocking objects (e.g., furniture) in the environment can block a photodiode, preventing it from capturing light rays from ceiling LEDs and recovering the virtual shadow map. *Second*, ceiling LED panels are commonly placed with intervals, not only for the cost, but also to circumvent other ceiling infrastructures like pipes, vents, and temperature sensors. The resulting layout of LED panels can be non-uniform and irregular. It presents a non-trivial constraint for the photodiode placement strategy to maximize the light rays captured by each photodiode. *Third*, photodiodes have their own limit on the viewing angle, i.e., Field of Vision (FoV). They cannot accurately perceive light rays coming near or outside its FoV. Thus, the photodiode's FoV is one additional constraint that the placement strategy has to consider given the layout of LED panels and the furniture.

User Mobility. Last but not least, user mobility brings additional challenges not only to the photodiode placement, but also to the reconstruction of the user skeleton. To the photodiode placement, user mobility means that the placement cannot be simply optimized towards a fixed user location, rather, it needs to ensure that photodiodes can capture sufficient light ray information to reconstruct user skeleton regardless of the user's current location and orientation. To the skeleton reconstruction, the uncertainty of user's location and orientation significantly increases the search space to identify the best-fit skeleton posture. As a result, it can lead to higher reconstruction latency and potentially higher reconstruction errors. We need efficient algorithms to guarantee real-time skeleton reconstruction while maintaining the reconstruction reliability.

We next present StarLight that is built atop the new sensing architecture (LED panels and few photodiodes) and addresses the above challenges. StarLight leverages a new light beacon design (§ 3) to deal with dense LEDs, a greedy algorithm to optimize the photodiode placement (§ 4) given the 3D layout of furniture and LED panels, and an efficient algorithm to track and reconstruct a moving user skeleton (§ 5).

3. RECOVERING VIRTUAL SHADOWS

We first describe how StarLight enables each photodiode to separate light rays from dense LEDs and recover the virtual shadow map that would have been projected on the ceiling if the photodiode were an LED. To support a large number of LEDs with a limited number of light beacon frequencies, StarLight reuses light beacon frequencies in the time domain while ensuring accurate and reliable separation of all light rays at each photodiode. We next present the new light beacon design for the LEDs, followed by the light ray separation and blockage detection schemes at photodiodes.

3.1 Time-Based Light Beacon

In the presence of dense LEDs, directly applying the prior light beacon design is no longer effective, mainly because it requires a large number of light beacon frequencies. As shown in the prior study [36], the highest light beacon frequency is limited by the sampling rate of the micro-controller fetching data from the photodiode, and the lowest has to be above 1 KHz to avoid the flickering effect [33, 35]. Thus, the more light beacon frequencies selected in this range, the smaller interval between adjacent frequencies. It results into faster transition between rising or falling edges in the light intensity perceived by the photodiode, which can outpace the photodiode's response time and lead to detection errors.

[4] Given the set F of light beacon frequencies, we have $T_{min} \leq (1/\max(F \setminus \{\max(F)\}) - 1/\max(F))/2$.

We address the problem by adding the time domain information to separate LEDs. The key idea is to reuse light beacon frequencies over time, so that it reduces the number of frequencies required to support a given number of LEDs. In particular, given N LEDs within a photodiode's viewing angle and the set F_b of available light beacon frequencies, we divide LEDs into groups, where each group contains no more than M (i.e., $|F_b|$) LEDs. Each LED i in the same group is assigned with a different light beacon frequency $f_i \in F_b$, and the same set F_b of frequencies are reused across groups. LED groups transmit light beacons in turns following a fixed order. When a group of LEDs are transmitting light beacons in the assigned beacon slot, LEDs in other groups flash at the *base frequency* $f_{base} \notin F_b$. This avoids two LEDs flashing at the same frequency simultaneously. Therefore, each LED i is identified by its light beacon frequency f_i and its beacon slot order b_i. This design echoes the principle of Pulse Position Modulation [13], where the time position of a pulse encodes bits. Similarly, here the time (beacon slot) when the light beacon occurs conveys additional information for the photodiode to identify light rays from an LED. The minimal number of beacon slots is $\lceil \frac{N}{M} \rceil$, where N is the number of LEDs within each photodiode sensor's FoV, and M is the maximal number of light frequencies that a photodiode can differentiate.[5] Thus, only the higher LED density needs more beacon slots. For a given LED density, we can scale the system to a large area without increasing beacon slots. We next describe two design enhancements to facilitate the light ray separation.

Synchronization. Adding the time domain information naturally requires tight synchronization among LEDs, as well as between LEDs and photodiodes, so that each LED transmits light beacons accurately in its assigned beacon slot and photodiodes correctly identify the start of each beacon slot. The synchronization among LEDs is straightforward, as LEDs are centrally controlled by a micro-controller[6]. The synchronization between LEDs and photodiodes is trickier. To that end, we add a *preamble slot* right before every set of beacon slots. In the preamble slot, all LEDs flash at the preamble frequency f_p (different from f_{base} and all light beacon frequencies F_b). The preamble creates a unique pattern for photodiodes to identify the start of following beacon slots. Figure 5 illustrates an example with two LEDs and two beacon slots, where these two LEDs are in different LED groups and thus transmit light beacons in different beacon slots.

Frequency Assignment. To further enhance the reliability of light beacon detection at photodiodes, we judiciously assign the light beacon frequencies F_b to LEDs based on their locations. The rationale is originated by our experimental finding: when light rays flashing at different frequencies arrive at a photodiode, the photodiode better extracts frequency powers of light rays with lower flashing frequencies. The reason is that the photodiode's response time results into a gradual rise or fall during light intensity changes. For light rays with lower flashing frequencies, more light signals are sampled, leading to higher frequency powers in the frequency domain and more robust blockage detection. Hence we assign lower frequencies to LEDs that are further from photodiodes, since their light rays arrive close to the photodiodes' FoV and their detections are less reliable. Specifically, we rank LEDs in a descend order based on their average distances to photodiodes that can perceive their light rays. We then assign to each LED a light beacon frequency from the lowest to the highest.

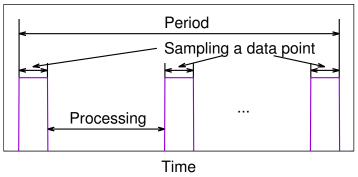


Figure 6: StarLight's processing pipeline. StarLight divides the whole data processing into small pieces and process these pieces during the intervals between adjacent samplings.

3.2 Blockage Detection

Given the light beacons from the LEDs, the micro-controller connecting to each photodiode performs two tasks: 1) *Sampling*: it periodically samples the light intensity data; and 2) *Processing*: it locates the preamble slot and thus the following beacon slots, fetches the light intensity values within each beacon slot, computes the frequency powers at the light beacon frequencies via FFT. The micro-controller transfers the results to a server. The server detects the blockage of a light ray based on its light beacon frequency power and its beacon slot number. It then recovers the virtual shadow map for this photodiode. We next discuss each step in detail.

Interleaving Sampling and Processing. To speed up the data sampling and processing at the micro-controller, we interleave the sampling and processing tasks, rather than executing them sequentially. As illustrated in Figure 6, during the intervals between adjacent data samplings, the micro-controller processes the data sampled in the last period. The interleaving is feasible because the maximal sampling rate (250 kHz) of the micro-controller's Analog-to-Digital Converter (ADC) is six times higher than the highest light beacon frequency (41.5 KHz, see § 6). Thus, the micro-controller can slightly space out its neighboring samplings and insert the processing task in between. The interleaving minimizes the idle duration of the micro-controller and fully utilizes its computation power. It is key for the system to later recover virtual shadow maps at a high frame rate.

Locating Preamble. Given the set of sampled data points D from one period, the micro-controller first locates the start of the preamble slot, which enables it to identify the subsequent beacon slots and compute frequency powers for each beacon slot. A strawman method is to slide a window with the preamble length L_p, perform FFT over the points in this window, and identify the window with the highest frequency power[7] at preamble frequency f_p. This method, however, requires performing FFT operations $|D|$ times, leading to a high processing delay.

To speed up the search, we locate the preamble in two steps. We first coarsely identify the region where the preamble potentially resides, and then narrow down to this region to conduct finer-grained search. Specifically, for the coarse-grain search, we divide D into disjoint subsets each with a length L_p of the preamble slot. Given the resulting $\frac{|D|}{L_p}$ subsets, we perform FFT over the points of each subset, extract the frequency power at f_p, and identify the subset $D_{i^\star}$ with the highest frequency power at f_p. Then the start of the actual preamble slot should lie in proximity of $i^\star$. In the second step, we fine-tune the $i^\star$ value using a binary search algorithm. The idea is to examine two candidate points within a range L of the current $i^\star$. We compute the frequency power at f_p using the

[5] M depends on the speed of the photodiode's rise/fall time and the ADC sampling speed of the mirco-controller it connects to.

[6] The FPGA board (Digilent Basys 3 FPGA board) used in our testbed (see § 6) introduces less than 1-ns clock shift.

[7] One can also compute cross-correlation [46] over the series of frequency powers of f_p to locate the start of the preamble. We find that locating the peak suffices and entails less computation.

Algorithm 1: Locating the preamble slot.

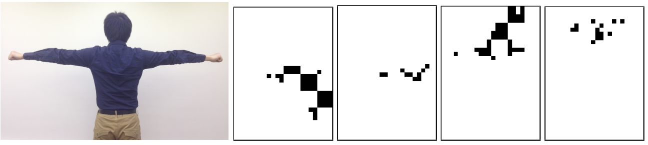
```
input  : 1) $L_p$, preamble slot length; 2) $f_p$, preamble frequency, i.e.,
         LED's flashing frequency in the preamble slot; 3) $D = \{d_i\}$,
         sampled light intensity data
output: $i^\star$, the start time of the preamble slot.

for $i \leftarrow 1$ to $\frac{|D|}{L_p}$ do
    $D_i = \{d_k | (i-1)L_p < k \leq i \cdot L_p\}$
    $p_i = \texttt{FFT}(D_i, f_p)$
end
$p^\star = \max(\{p_i | i \in [1, \frac{|D|}{L_p}]\})$
$i^\star = \text{argmax}_{i \in [1, \frac{|D|}{L_p}]}\, p_i$
// Binary search to refine $i^\star$
$L = L_p$
while $\frac{L}{2} \geq 0$ do
    $L = \frac{L}{2}$
    $D_l = \{d_i | i \in [i^\star - L, i^\star - L + L_p)\}$
    $D_r = \{d_i | i \in [i^\star + L, i^\star + L + L_p)\}$
    $p_l = \texttt{FFT}(D_l, f_p)$
    $p_r = \texttt{FFT}(D_r, f_p)$
    if $p_l > p^\star$ then
        $p^\star = p_l$
        $i^\star = i^\star - L$
    end
    else if $p_r > p^\star$ then
        $p^\star = p_r$
        $i^\star = i^\star + L$
    end
end
```

L_p points starting from each candidate point, and determine the next search direction. We gradually shrink the search range L and the algorithm converges to the final output $i^\star$. Algorithm 1 lists the specific steps, where $FFT(D_i, f)$ applies FFT over the data points in D_i and extracts the frequency power at f. The total number of FFT operations in Algorithm 1 is $(\frac{|D|}{L_p} + 2 \cdot \log_2 L_p)$.

Adaptive Blockage Detection. The micro-controller sends to the server the extracted frequency powers at light beacon frequencies of all beacon slots. The server then associates each frequency power with an LED based on the light beacon frequency and its beacon slot. Since the frequency power is directly proportional to the light intensity, we can detect the blockage of an LED by examining its frequency power change. Given the heterogeneous light intensity from different LEDs, we compute the normalized, rather than the absolute, frequency power change as $\Delta P_{ij}(t) = |\frac{P_{ij}^{\text{nonBlock}} - P_{ij}(t)}{P_{ij}^{\text{nonBlock}}}|$, where P_{ij}^{nonBlock} and $P_{ij}(t)$ are the average non-blocking frequency power, and the frequency power at time t, from LED i at photodiode j, respectively. The light ray is considered to be blocked only if $\Delta P_{ij}(t)$ is above a threshold δ_{ij}. Our experiments show that light rays with higher intensity values experience larger normalized frequency power change caused by blockage. Therefore, unlike the prior design [36] that uses a uniform threshold for all light rays, StarLight sets δ_{ij} based on the light intensity I_{ij} normalized to the maximal light intensity I_{max} among all light rays. The light propagation model [30] indicates $I_{ij} \propto \frac{\cos\theta_{ij}}{d_{ij}^2}$, where θ_{ij} is the incidence angle, and d_{ij} is the distance between LED i and photodiode j. Thus, we compute δ_{ij} as:

$$\delta_{ij} = P_{min} + (P_{max} - P_{min}) \cdot \frac{\cos\theta_{ij}}{d_{ij}^2} \cdot \frac{d_{min}^2}{\cos\theta_{min}}, \qquad (1)$$

where P_{min} and P_{max} are the minimal and maximal normalized frequency power changes, respectively, d_{min} and θ_{min} are related to I_{max} and are the minimal distance and incidence angle across all LED-photodiode pairs. We set P_{max} as 0.7 and P_{min} as 0.4 based on experiment results.



(a) User posture (b)Virtual shadow maps at 4 photodiodes

Figure 7: Four example virtual shadow maps recovered from our StarLight testbed, under a simple user posture.

After detecting the blockage between each LED and a photodiode j, we recover the virtual shadow map $\widetilde{S}_j(t)$ at this photodiode's view point. With N LEDs, we have $\widetilde{S}_j(t) = \{\widetilde{s}_{ij}(t) | i \in [1, N]\}$, where $\widetilde{s}_{ij}(t) = 1$ if $\Delta P_{ij}(t) \geq \delta_{ij}$, and $\widetilde{s}_{ij}(t) = 0$ otherwise. Figure 7(b) shows examples of the virtual shadow maps at five photodiodes, given the user posture in Figure 7(a). The gaps in the virtual shadow maps are due to the non-uniform layout of the LED panels (Figure 8(b)). Clearly, the information conveyed by a virtual shadow map largely depends on the photodiode's location. Next we discuss how StarLight optimizes the photodiode placement to facilitate the user skeleton reconstruction.

4. OPTIMIZING SENSOR PLACEMENT

StarLight judiciously places photodiodes given a 3D environmental setup (e.g., layout of furniture and LED panels, room size and height). We aim to ensure that the virtual shadow maps at these photodiodes best facilitate the later skeleton reconstruction, considering furniture blockage, photodiode's limited FoV, and user mobility. A simple brute-force method is to exhaustively examine each possible combination of photodiode locations, estimate the resulting reconstruction errors, and identify the optimal photodiode placement that leads to the minimal reconstruction error. Clearly this method is not scalable given the exponential search space.

To boost the efficiency, StarLight defines an effective metric to evaluate the efficacy of a placement instance, without the need to repeatedly run the reconstruction algorithm. A key advantage of using this metric is that it turns the placement optimization problem into a *submodular maximization subject to a cardinality constraint* while retaining its direct link to the reconstruction performance. Thus a greedy algorithm to place photodiodes is highly effective. The algorithm entails polynomial-time complexity and achieves a proven $(1 - 1/\text{e})$ approximation [44] to the optimal. Next, we first formally define the problem and analyze its property. We then present the greedy placement strategy in detail.

Problem Statement. While rich literature [10, 71] has studied the optimization of sensor placement in sensor networks, prior techniques have focused on information theoretic metrics [31]. However, in our context, the optimization goal is minimizing the 3D skeleton reconstruction error, which requires a new objective function. To eliminate the need to involve the skeleton reconstruction algorithm, we define an intermediate metric to evaluate the efficacy of a placement instance in minimizing reconstruction errors. The metric is based on a basic principle of the skeleton reconstruction, which leverages the information on which light rays intersect the user body to infer the 3D skeleton. Thus, to facilitate skeleton reconstruction and support user mobility, light rays arriving at the photodiodes should be diversely spread out, maximizing the likelihood that photodiodes capture light rays intersecting the user body.

Algorithm 2: Greedy placement of photodiodes.

```
input : 1) $U = \{u_1, ..., u_N\}$, $N$ LEDs' locations; 2) $\Omega$, feasible
        locations to place photodiodes; 3) $Z$, cube's edge length; 4)
        $E$, environmental setup (furniture layout, room size and
        height);
output: $w_1^\star, ..., w_K^\star$, locations of all $K$ photodiodes.
$C$ = genCubes($E, Z$)// divide space into cubes
for $m \leftarrow 1$ to $|C|$ do
    $cross[c_m] = 0$ // clear cube status
end
for $i \leftarrow 1$ to $K$ do
    $new\_cubes = \emptyset$
    for $w_j \in \Omega$ do
        $tmp = \emptyset$
        for $k \leftarrow 1$ to $N$ do
            if furBlock($E, \overrightarrow{u_k w_j}$) or outFoV($u_k, w_j$) then
                continue
            end
            for $m \leftarrow 1$ to $|C|$ do
                if $cross[c_m] = 1$ then continue
                if dist($\overrightarrow{u_k w_j}, c_m$) $< \frac{Z}{2}$ then
                    $tmp = tmp \cup \{c_m\}$
                end
            end
        end
        if $|new\_cubes| < |tmp|$ then
            $new\_cubes = tmp$
            $w_i^\star = w_j$
        end
    end
    for $c_m \in new\_cubes$ do
        $cross[c_m] = 1$// mark newly-crossed cubes
    end
    $\Omega = \Omega \setminus \{w_i^\star\}$
end
```

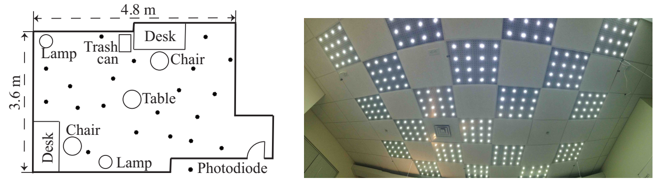
Based on this rationale, for a given placement instance $W = \{w_1, ..., w_K\}$ of K photodiodes, we evaluate its efficacy using the volume of the 3D space intersected by the light rays arriving at these photodiode locations. To simplify the calculation, we discretize the 3D space into small uniform cubes[8]. We define the metric $A(W)$ as the number of cubes crossed by light rays arriving at locations in W, ruling out light rays blocked by furniture or outside the photodiode's FoV. The placement optimization problem is to identify the placement $W^\star$ that maximizes this metric. Let Ω denote all the feasible locations to place photodiodes avoiding furniture, and $Q(w)$ is the set of cubes crossed by light rays arriving at location w. The problem can be formalized as a maximum coverage problem with a cardinality constraint:

$$\text{Maximize} \qquad A(W) = |\bigcup_{w \in W} Q(w)| \tag{2a}$$

$$\text{subj. to:} \qquad W \subset \Omega \tag{2b}$$

$$|W| \leq K \tag{2c}$$

The maximum coverage problem is NP-hard, but its objective function is known to be monotonic and submodular [20]. In our context, $A(W)$ increases monotonically with more photodiodes, and the contribution of the subsequent photodiodide diminishes. A greedy solution hence yields a $(1 - 1/\mathrm{e})$ approximation ratio, essentially the best approximation ratio one can achieve [16].



(a) Photodiode placement (b) LED panel layout

Figure 8: Placement of 20 photodiodes and 16 LED panels in StarLight testbed (§ 6).

Greedy Placement Strategy. Our greedy placement strategy works as follows. Given an environmental setup (e.g., furniture layout, room size and height), we can derive the free space E that user's body interacts with. We divide E into a set of cubes C (genCubes($\cdot$)), where the edge length of each cube $c_m \in C$ is Z. We place K photodiodes sequentially. To determine the location of the i-th photodiode, we examine each candidate location $w_j \in \Omega$ and mark the cubes crossed by light rays from all LEDs to w_j. We exclude the light rays already blocked by the furniture (furBlock($\cdot$)) or outside the photodiode's FoV (outFoV($\cdot$)). We then count the number of newly-crossed cubes that have not been crossed by light rays arriving at the previous $(i-1)$ photodiodes. We identify the location $w_i^\star$ that brings the most newly-crossed cubes and place the i-th photodiode there. Algorithm 2 lists the details, where dist($\overrightarrow{u_i w_j}, c_m$) computes the distance between a vector $\overrightarrow{u_i w_j}$ and the center of cube c_m. Figure 8(a) marks the photodiode locations output by our algorithm for a given furniture setup and the LED panel layout in our testbed (Figure 8(b)).

The greedy algorithm runs in polynomial-time complexity. It outputs the optimized photodiode locations within 1 minute for our current setting. It implies that upon any environment changes (e.g., furniture removal or addition), we can quickly update the photodiode locations by running the placement algorithm again. In the future, to further speed up the calculation, we plan to examine algorithms that update the locations of photodiodes only within the areas affected by the environment change, rather than computing the locations of all photodiode sensors.

5. TRACKING A MOVING SKELETON

Armed with continuous streams of virtual shadow maps from different points of view (photodiodes), finally StarLight localizes and infers the user 3D skeleton over time. Unlike the prior work [36] that reconstructs the body skeleton of a static user with known orientation, StarLight seeks to reconstruct the skeleton of a moving user with unknown location and orientation while maintaining a similar reconstruction accuracy. Next, we begin with an overview of the inference algorithm design and then describe the key components in detail.

Overview. Reconstructing a moving skeleton is much more challenging because the search space of user's body segments expands exponentially given the uncertainty of user location and orientation. To address this challenge, we aggregate the virtual shadow maps from all photodiode sensors and extract coarse body features from the aggregated shadow map. These body features roughly indicate the user's location and possible orientation. Thus, they can be leveraged to reduce the search space greatly. We also leverage user's movement continuity to further reduce the search space and fine-tune the reconstruction results. Since the stream of virtual shadow maps are generated at a high frame rate (40 Hz, see § 7.2), the location and orientation offset between two adjacent inferences is very small. Hence the inference algorithm can start from the

[8] For accurate tracking, each cube needs to be no larger than the smallest body part to be tracked. In our current implementation, we set the cube's edge length to 10 cm to allow the system to track user hands.

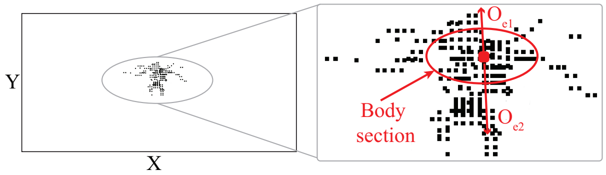


Figure 9: An example aggregated shadow map for the user posture in Figure 7(a). The ellipse maps to the user body and the ellipse minor axis (two arrows O_{e_1} and O_{e_2}) is the potential user orientation.

previous reconstruction results, search only within their small local range, and identify the optimal skeleton posture.

In particular, the inference is an iterative procedure with the following steps: 1) it refines the current set of virtual shadow maps by fixing apparent erroneous pixels; 2) it aggregates the shadow maps from all photodiodes and extracts coarse body features; 3) it then estimates the user's 2D location based on the extracted body feature and the last inferred skeleton posture; 4) next it computes user's current orientation based on the extracted body feature and user's moving trajectory; 5) using the inferred 2D user location and user orientation, it searches for the skeleton posture that best matches observed virtual shadows; 6) it repeats the above steps and iteratively refines the inference; and 7) finally it applies a Kalman Filter to smooth the inference results. Step 5) - 7) are based on the inference algorithm in the prior work [36]. Thus, we omit their details and focus on introducing the other steps.

Refining Virtual Shadow Maps. We start with correcting the apparent pixel errors in the collected virtual shadow maps. These erroneous pixels are caused by blockage detection errors on certain pairs of LED and photodiode (e.g., LED's light ray coming close to the photodiode's FoV). We spot these errors based on simple principles as follows: our body consists of connected body parts, thus the resulting shadow silhouette cannot have disconnected components; within a short time span between two consecutive set of virtual shadows, the offset generated by body movement is very small given the movement continuity. Applying these principles, we first pre-process each single virtual shadow map by filtering out black pixels disconnected from user's main body and far away from the previous user location. Next we leverage the past k (20 in our implementation) virtual shadow maps from the same photodiode and smooth pixels in the current virtual shadow map based on movement continuity. Such preprocessing prevents the propagation of shadow map errors to the later skeleton reconstruction.

Extracting Coarse Body Features. Based on the refined virtual shadow maps, we now extract coarse-grained body features from these maps. To support diverse users, we seek common body features independent of user's postures and body details. Specifically, we choose the cross section of a user's main body as the main body feature, as it is universally an ellipse regardless of user's postures. However, even for such a simple body feature, its extraction is still challenging in our context, because these virtual shadow maps (see Figure 7(b)) can be incomplete (limited by the photodiode's viewing angle) and they contain extremely low number (320) of pixels compared to typical photos or images. We overcome this challenge by aggregating the blockage information collected by all photodiodes. For all the blocked light rays observed by these photodiodes, we consider the intersection of these light rays with a flat horizontal plane at the height of the user's waist. Aggregating these intersection points leads to an aggregated shadow map projected on this horizontal plane. It reveals the cross section of the user's main body as an ellipse. To localize the ellipse, we exhaustively search the 2D area near the centroid of the aggregated shadow map and identify the ellipse that covers the most intersection points. The ellipse center approximates user's location and the direction of its minor axis indicates user's potential orientation. Figure 9 shows an example of the aggregated shadow map for the user posture in Figure 7(a), where we mark the identified ellipse representing user's main body and its minor axis indicating user orientation.

Locating A Moving Skeleton. Next, we locate the center of the user's main body in the 2D plane at time t. Although the center of the extracted ellipse provides an estimate of the user's location, its location error can be up to 10 cm, too high for inferring the skeleton posture accurately later. To lower the location error, we leverage user's movement continuity to refine the location estimate. Because of the short time interval ($<$ 25 ms) between adjacent posture inferences, the skeleton posture at time t shares a high similarity with the last inferred posture at time $(t-1)$. Therefore, we can leverage the last inferred body segments B_{t-1} to estimate the most likely location offset at time t. In particular, we define a fitness function $F(\Delta x_t)$ to evaluate the likelihood of a candidate offset Δx_t. Based on current virtual shadow maps, $F(\Delta x_t)$ calculates the summation of the minimal distances from the blocked light rays to the body segments after applying the offset Δx_t to the last inferred posture B_{t-1}. We then estimate the user's current offset $\Delta x_t^\star$ as the one with the minimal $F(\Delta x_t)$ value. With N LEDs and K photodiodes, $\Delta x_t^\star$ is written as:

$$\Delta x_t^\star = \operatorname*{argmin}_{\Delta x_t} F(\Delta x_t), \text{ where}$$

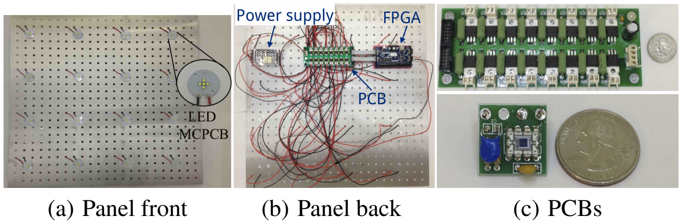
$$F(\Delta x_t) = \sum_{\substack{\tilde{s}_{ij}(t)=1 \\ i\in[1,N], j\in[1,K]}} \min_{b_m \in B_{t-1}} (\mathtt{dist}(l_{ij}, b_m + \Delta x_t) - r_m), \quad (3)$$

$\mathtt{dist}(l_1, l_2)$ calculates the perpendicular distance between two line segments l_1 and l_2, l_{ij} is the line segment between LED i and PD j, b_m is the 3D vector of body segment m, r_m is the radius of body segment m. Guided by the extracted ellipse, we search for $\Delta x_t^\star$ only within the 10-cm circle centered on the ellipse center, which greatly shortens the search duration. The user's current location x_t then is $(x_{t-1} + \Delta x_t^\star)$.

Determining User's Orientation. Given the direction of the ellipse's minor axis, user's orientation O_t at time t is within $\mathbb{O} = [O_{e1} - 30°, O_{e1} + 30°] \cup [O_{e2} - 30°, O_{e2} + 30°]$. We further reduce the search space by leveraging the movement continuity. Specifically we set O_t as follows. If the user moves in the 2D plane during [$t-1$, t], i.e., $\Delta x_t^\star \neq 0$, we set O_t as $\overrightarrow{x_{t-1}x_t}$, where x_{t-1} and x_t denote the 2D location of the center of user's main body at time $t-1$ and t, respectively. Otherwise, if the user stays at the same 2D location within [$t-1$, t], i.e., $\Delta x_t^\star = 0$, we analyze the mean O' of the previous k (5 in our implementation) orientation vectors. O_t is then within the intersection of $\mathbb{O}$ and $[O' - 30°, O' + 30°]$. We then rotate the user's 3D skeleton within the search space, compute a fitness value (similar to Eq. (3)) to evaluate how well it matches current virtual shadow maps, and search for the best-fit user orientation.

6. SYSTEM IMPLEMENTATION

We build and deploy StarLight in an office room (3.6 m× 4.8 m) with 2.6-m ceiling height. We mount 20 customized LED panels in 15-cm interval on the ceiling (Figure 8(b)). We distribute 20 low-cost photodiodes (OPT101 [4]) on the floor based on the greedy placement strategy (§ 4). Each photodiode connects to an Arduino Due micro-controller. A server collects data from all Arduino boards, runs our skeleton reconstruction algorithm (§ 5), and visualizes the reconstructed skeleton on the monitor (Figure 1).



(a) Panel front (b) Panel back (c) PCBs

Figure 10: (a)(b) A customized LED panel and (c) the PCBs for the LED panel (top, 16 cm × 6.4 cm in size) and for the photodiode (bottom, 2 cm × 2 cm in size)).

LED Panels. We design and fabricate 20 LED panels (61 cm × 61 cm in size), powered by independent 15-V DC supply. We do not directly use off-the-shelf LED panels because they commonly connect all LED chips in series. Thus, they do not allow independent control of each LED, which, however, is required for implementing StarLight's light beacons. Each panel contains 16 off-the-shelf LED Metal-Core PCBs (MCPCB) connected in parallel. Each MCPCB functions as a point light source with 5 LED chips (CREE XPG2) clustered in the center (Figure 10(a)). To control each LED MCPCB independently, we fabricate the electrical components as a PCB (Figure 10(c) top) connecting to each panel. We then connect the PCB to an FPGA board (Digilent Basys3), where we implement our light beacon design and synchronization scheme (§ 3.1). We leverage two 8-pin connectors on the FPGA to control LED's flashing frequency. To synchronize the FPGAs of all panels, a separate FPGA board sends a pulse to all 20 FPGAs every 20.8 ms.

We implement light beacons with 3 beacon slots and 108 flashing frequencies (107 for light beacons and 1 for f_{base}), ranging from 20.8 kHz to 41.5 kHz. Three beacon slots last 19.3 ms and the synchronization slot lasts 1.5 ms. The resulting illuminance in the room center is 400 lx on average and 500 lx at maximum, meeting the indoor lighting standard [51]. Figure 10(b) shows circuit boards, power supply, and wires on the back of the panel.

Photodiodes. For a miniaturized look, we design and fabricate a PCB (Figure 10(c) bottom) to include the photodiode and electrical components (e.g., resistors, capacitor). We then connect the PCB to an Arduino Due that samples the light signals from a photodiode. We select OPT101 as the photodiode for three reasons. First, it has a high responsivity (0.45 A/W for visible light wavelength of 650 nm), which helps detect multiple concurrent flashing frequencies. Second, it supports a wide FoV (140° on x-axis and 100° on y-axis based on our measurements), which enables the photodiode to perceive almost all LEDs in our testbed. Third, we can adjust its bandwidth of the frequency response to 56 kHz[9], which is much higher than the highest flashing frequency (41.5 kHz).

Micro-controllers. We implement our processing pipeline and blockage detection scheme (§ 3.2) on Arduino Due boards. The ADC sampling rate is set to 83.3 kHz and the number of sample points for a virtual shadow map is 1736. Due to the electrical noise on the PCB and control boards, the mean synchronization offset is 3 frames on average and 12 frame at the maximum. Thus, we set the beacon slot longer than the preamble slot to minimize the impact of the synchronization offset. We apply a 128-point FFT to detect the preamble slot and a 512-point FFT to extract light beacon frequency powers in the beacon slots. We implement our processing pipeline by adding interrupts in the main thread.

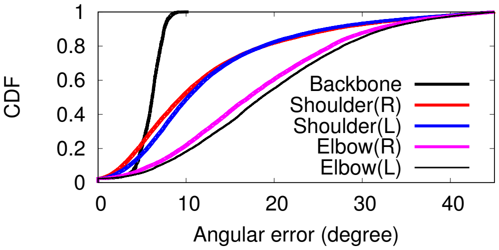


Figure 11: The overall reconstruction accuracy of StarLight, aggregating 195304 skeleton reconstruction results.

Arduino boards compute light beacon frequency powers of the beacon slots and send them to the server using RS232 ports, whose buad rate is limited to 115.2 Kbps. To improve the transmission efficiency, we increase the RingBuffer on the Arduino board to 4 KB and compress the data packet by rescaling the frequency power to 16. Thus, the packet length for a virtual shadow map is 72 bytes.

Server. The server aggregates data packets from all Arduino boards, recovers the virtual shadow maps, and reconstructs the user skeleton in the 3D space. To speed up the whole process, we implement virtual shadow recovering and skeleton reconstruction in C++ as two separate threads. Due to the RS232 transmission constraint, we place the server less than 3 m away from all Arduino boards.

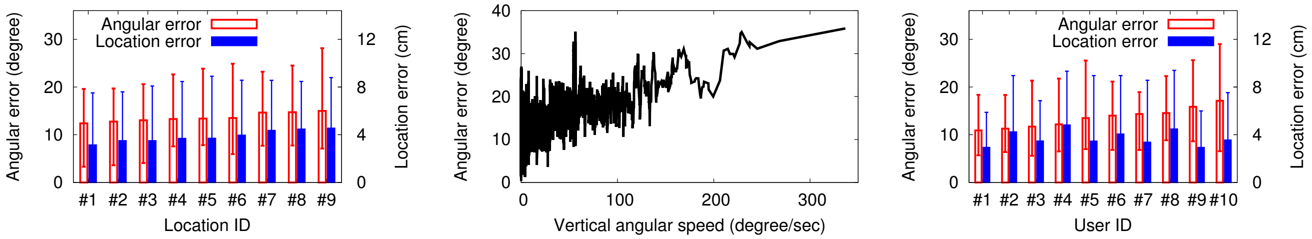
7. StarLight EVALUATION

We now examine StarLight's practical performance using our testbed. We aim to evaluate its reconstruction accuracy and latency, the efficacy of each design component, and the impact of the environmental setting such as ambient light.

Experimental Setup. We evaluate StarLight in a typical indoor setting with furniture (e.g., tables, chairs, floor lamps, see Figure 8(a)). The only calibration required by StarLight is for each photodiode to measure the light intensity value without any users present, which helps configure the $P_{ij}^{nonBlock}$ for blockage detection (§ 3.2). Since $P_{ij}^{nonBlock}$ is the frequency power and StarLight's light beacon frequency (20 KHz – 40 KHz) is much higher than that of the ambient light fluctuation, we do not need to repeat the calibration under different ambient light conditions (see § 7.4 that examines the impact of ambient light). For a given furniture layout, the calibration only needs to be performed once. In each experiment, the user walks around in the room and freely performs consecutive upper-body gestures as he/she wishes, rather than performing specified gestures. The example gestures we have collected include pointing, waving, circling, stretching and more. Each experiment lasts 14 - 35 minutes and we repeat the experiment by 5 rounds. By default, we place 20 photodiodes at locations calculated by the greedy strategy (§ 4).

Ground Truth. In parallel to operating StarLight, we use a Kinect sensor (Version 2) to collect the 3D coordinates of user's key body joints continuously and treat Kinect data as the reference to evaluate StarLight's accuracy. We choose Kinect for two reasons. *First*, except Kinect, the only other existing system with the comparable ability is VICON. However, the room size requirement of installing VICON makes it infeasible in our experiment room. *Second*, compared to human labeling, which is the standard technique to obtain ground truth in computer vision [25, 42], the latest Kinect provides real-time, fine-grained, continuous (30 FPS) reconstruction results of all body joints, greatly facilitating the comparison to StarLight's reconstruction results. We do recognize, however, Kinect's imperfection, i.e., the limited viewing angle and support-

[9] We change the photodiode's bandwidth by connecting two 100-$k\Omega$ resistors and one 56-pF capacitor to the photodiode.



(a) Impact of location error

(b) Impact of gesture speed

(c) Impact of user diversity

Figure 12: Analyzing how StarLight's reconstruction accuracy is affected by the 2D localization error, gesture characteristics, and user diversity.

ing distance. Thus, we thoroughly test Kinect tracking users located in varying angles and distances, compare its results to human labeling results, and identify Kinect's best working condition. To acquire the human labeling results, we mount three DSLR cameras on the sides of the testbed. We build a MATLAB-based tool to label the key body joints in the captured videos and acquire the body skeleton data in the 3D space. We observe that as long as users are within 45° and 2 m - 6 m to the center of Kinect, the mean angular offset between Kinect and the human labeling method is negligible ($< 3°$). Given that Kinect can acquire body postures in real time while human labeling can take days to finish, we leverage Kinect data collected within its working range as the ground truth.

7.1 Reconstruction Accuracy

We begin with examining StarLight's overall reconstruction accuracy with ten users (see Table 1)[10]. We evaluate the reconstruction accuracy by calculating the absolute angular difference between the body segment vectors inferred by StarLight and that by Kinect. We focus on five key body joints including the backbone, left and right shoulder joints, left and right elbow joints. This angular metric has been widely used by prior studies [6, 9, 32, 36, 43] to examine skeleton tracking and reconstruction performance.

Figure 11 plots the CDFs of angular errors for the five body joints. Overall, StarLight achieves 5° mean angular error for the backbone joint and 14° mean angular error for all joints of the participants. The error is slightly higher than the prior design LiSense (10° [36]), however, StarLight addresses multiple practical constraints (e.g., furniture blockage, user mobility) while maintaining a similar accuracy. In comparison, LiSense assumes a static user with known location and orientation, which simplifies the reconstruction problem and eliminates the impact of user location tracking error on the reconstruction performance. Next we dive into specific factors that can affect StarLight's accuracy.

User Mobility. While the support of user mobility is one of the key features differentiating StarLight from LiSense, any slight inaccuracy in localization also contributes to the reconstruction error. Here we take the center of a user's main body projected on the floor as the user's location. To gather user's location data accurately during experiments, each participant wears a laser pen on the chest such that the laser pen points to the floor and forms a laster dot representing user's body center projected on the floor. We record the laser dot position to acquire the precise ground truth of the user location. To examine how the localization error is correlated with the reconstruction accuracy, we plot both errors (mean reconstruction errors for all body joints, and localization error) for 9 marked locations in Figure 12(a). We also include error bars to cover the minimal and maximal errors. As expected, we observe a positive correlation between these two errors. In particular, compared with the minimal localization error (3.14 cm Euclidean offset at Location 1), the highest location error (4.55 cm at Location 9) leads to 2.5 ° higher angular error. The reason is that StarLight relies on the inferred user location and orientation to identify the best-fit 3D skeleton posture. 2D localization error not only enlarges the search space for the reconstruction algorithm, but also misleads the algorithm to a wrong local minimum.

Table 1: The body parameters of 10 participants.

User ID	1	2	3	4	5	6	7	8	9	10
Height (cm)	190	190	183	173	170	170	176	170	169	158
Weight (kg)	84	80	64	67	65	68	66	63	60	50

Gesture Characteristics. To examine the impact of user's gesture characteristics on StarLight's reconstruction accuracy, we aggregate the stream of body segment coordinates from Kinect, calculate the horizon and vertical offsets between adjacent body segment coordinates, and examine their correlation with reconstruction accuracy. Our key finding is that the speed of vertical movement has a noticeable impact on the accuracy (Figure 12(b)), while the impact of horizon movement speed does not exhibit noticeable patterns. The reason is that in the current StarLight testbed, all light rays are emitted from the ceiling to the floor and vertical movement affects fewer light rays than horizontal movements. To better deal with fast vertical movements, one solution is to consider light sources such as floor lamps or wall lights, which emit light rays from the sides.

User Diversity. We also compare StarLight's performance across users, seeking to examine the impact of user diversity on the reconstruction accuracy. Figure 12(c) shows the the mean angular errors of all body joints for each individual participant, as well as the 5% and 95%-percentiles. Overall, the mean angular errors across users differ within 5°. Interestingly, the system achieves higher reconstruction accuracy for users with larger bodies. As an example, the mean angular error is only 10° for the user 1.9 m in height and 84 kg in weight. This is because larger bodies block more light rays and generate richer blockage information for analysis. As a result, the fitting function (Eq. (3)) can converge more quickly to identify the local minimum and lead to higher reconstruction accuracy.

7.2 Reconstruction Latency

We next examine StarLight's reconstruction latency. We focus on the latency of its two real-time components: the acquisition of virtual shadow maps (§ 3) and the skeleton reconstruction (§ 5), which determine the frames per second for displaying the reconstructed skeleton. The photodiode placement strategy runs offline.

Acquisition of Virtual Shadow Maps. Recovering the virtual shadow maps comprises four steps: 1) each Arduino Due samples signals from the photodiode; 2) it detects the preamble slot within the samples and locate the subsequent beacon slots; 3) then it fetches the sampled data of each beacon slots and extracts the

[10] Our study is conducted under the IRB approval at the local institution.

Table 2: Processing time of generating virtual shadow maps.

Step	ADC sampling	Preamble detection	Blockage detection	Transmission
Time (ms)	6.7	8.2	4.5	0.5

Table 3: Processing time of reconstructing a mobile skeleton.

Step	Localization	Orientation detection	Inference
Time (ms)	6.4 – 8.2	0.4 – 1.1	4 – 15.6

frequency power at light beacon frequencies; and 4) finally it sends the frequency powers to the server. The server detects the blockage of each light ray and recovers the binary virtual shadow maps.

Table 2 lists the latency of each step running at the Arduino Due board. Given the 83.3 kHz ADC sampling rate and 1736 samples for each shadow map frame, the total sampling duration is 20 ms. To speed up the ADC processing time, we modify the ADC register on the Arduino board to shrink the ADC processing time to 4 μs/sample. The processing time for 1736 points is 6.6 ms. To detect the preamble (Algorithm 1), we need to compute 128-points FFT 27 times and each FFT takes 300 μs. For blockage detection, we need to compute 512-points FFT 3 times and each FFT takes 1.3 ms, generating 324 bits to represent a single virtual shadow map. We enlarge the ring buffer on the Arduino board to 8 Kb in order to continuous transit data without further latency. Overall, StarLight generates concurrent virtual shadow maps within 20 ms.

Reconstruction of Mobile Skeletons. Inferring a user gesture is an iterative process, where the algorithm iteratively updates the user's location, orientation, and gestures to fit the captured virtual shadow maps. Table 3 lists the latency of the three key components in the reconstruction algorithm. In terms of the size of the shadow on the virtual shadow maps, the reconstruction duration varies from 10.8 ms to 24.9 ms. Figure 13 shows the reconstruction latency as the number of photodiodes varies. The reconstruction latency increases almost linearly as the number of photodiodes increases, since the algorithm processes more virtual shadow maps.

Overall, since the above two components run in parallel, StarLight can produce a reconstructed human skeleton within 25 ms. Thus, it can produce at least 40 user skeleton postures per second on the fly, faster than the frame rate (30 FPS) of the latest Kinect.

7.3 Microbenchmarks

We now move on to examining the efficacy of StarLight's key components. In particular, photodiode placement strategy and indoor localization play crucial roles for StarLight to reconstruct the 3D skeleton accurately. We conduct two additional experiments to evaluate these two key components.

Efficacy of Photodiode Placement Strategy. To evaluate the greedy placement strategy, we consider the uniform placement strategy as a baseline. In particular, to deal with the irregular shape of the space available for placing photodiodes, we use the Lloyd's algorithm [37], a well-known sensor placement algorithm, to uniformly place photodiodes in areas not covered by furniture. Figure 14(a) compares these two strategies in terms of reconstruction angular errors. Overall, the greedy placement achieves moderate improvements, with a shorter tail and 3° lower mean angular error. The main reason is that the uniform strategy considers only the 2D plane (i.e, the floor) information. Thus, it fails to take into account the furniture 3D structure and LED panel layout in the 3rd dimension. As a result, placing photodiodes at its calculated locations leads to lower reconstruction accuracy.

Localization Accuracy. To evaluate the 2D localization accuracy of StarLight, we conduct experiments with the participant standing at 30 marked spots. As described in § 7.1, the participant wears a laser pen on the chest and the laster pen points to floor. At each marked spot, the participant adjusts his/her location to ensure that the resulting laser point aligns with the marked spot. We then compare the actual locations of these marked spots and the locations inferred by StarLight to examine StarLight's localization accuracy. We repeat the experiment under different number of photodiodes and plot the CDF of localization errors in Figure 14(b). We observe that in the default setting (20 photodiodes) the median location error is 4 cm and the 95%-percentile tail is 9 cm. The error slightly increases with fewer photodiodes since less blockage information is available. Increasing the photodiodes to 25 brings diminishing returns. Given the possible offset introduced by the use of laser pen, we also compare the inferred user movement trajectory to that from Kinect. We observe that the mean trajectory deviation against Kinect is 2 cm and the 95%-percentile is 6 cm. Overall, StarLight's 2D location accuracy outperforms the latest localization techniques [33, 35, 50, 70] using VLC. StarLight fundamentally differs in that it does not require users to carry or wear any devices. The high 2D localization accuracy in StarLight is essential for its 3D skeleton reconstruction.

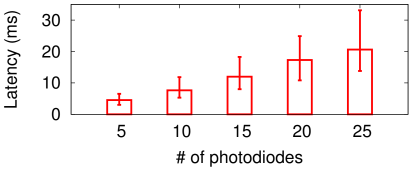
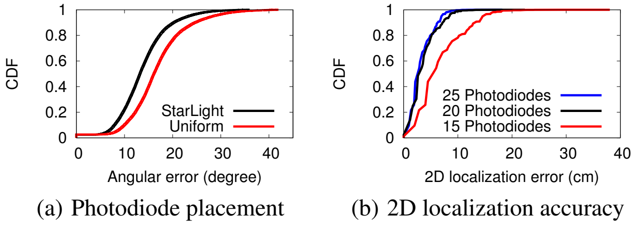


Figure 13: The impact of the number of photodiodes on StarLight's reconstruction latency.

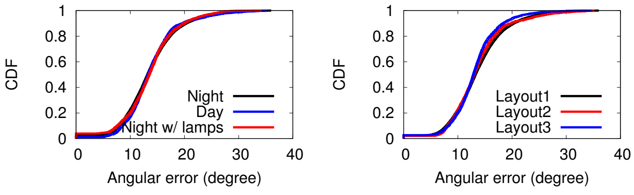


(a) Photodiode placement (b) 2D localization accuracy

Figure 14: Evaluating StarLight's photodiode placement and 2D skeleton localization.

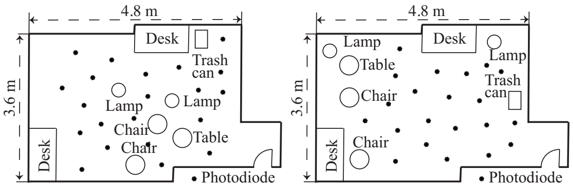
7.4 Environmental Factors

Impact of Ambient Light. First, we evaluate StarLight's reconstruction performance under natural ambient light varying from 300 lx at daytime, 70 lx at night with lamps, to 0 lx at night. We repeat the experiments in § 7.1 under the three ambient light conditions and plot the angular errors in Figure 15(a). Overall StarLight's performance is stable in all ambient light conditions. The difference in mean angular errors among the ambient light conditions is less than 1°. This is because StarLight's light beacon frequencies range from 20 KHz to 40 KHz, much higher than the blinking frequencies of other potential light sources, such as fluorescent light (60/120 Hz) and sun light (close to 0 Hz). The light from these interfering light sources can be easily filtered out in the frequency domain. Thus StarLight is robust against other interfering light sources. The reconstruction accuracy under bright ambient light (300 lx) is slightly lower than that without ambient light because high ambient light intensity pushes the photodiode into its saturation region and reduces its responsiveness. As the user blocks the LED, the corresponding frequency power change is smaller, leading to higher errors in virtual shadow maps and reconstruction results.



(a) Ambient light (b) Furniture layout

Figure 15: The impact of environmental factors such as ambient light (a) and furniture layout (b). In (b), layout1 is the default shown in Figure 8(a), while layout2 and layout3 are shown in Figure 16.



(a) Furniture layout2 (b) Furniture layout3

Figure 16: Two additional furniture layouts tested in Figure 15(b) as well as the optimized sensor placement.

Impact of Furniture Layout. Finally, we evaluate StarLight's reconstruction performance under different furniture layouts. We test two additional layouts (Figure 16) and plot the reconstruction accuracy in Figure 15(b). Overall the furniture rearrangement has negligible impact on StarLight. StarLight performs slightly better in layout3 because most furniture in this layout are clustered, leading to a larger sensing area and thus more candidate locations to place the photodiodes. As a result, the sensor placement strategy identifies better locations for the photodiodes to collect the light blockage information and improves the reconstruction accuracy.

8. RELATED WORK

We categorize related work into two classes:

Sensing with On-Body Devices. The first category of work relies on on-body devices (e.g., smartphones, wrist bands) to sense our behaviors and activities. In particular, prior studies in [34, 39, 38] have examined the use of smartphone sensing data to infer user's social events, activities, wellness, or even the psychological states. In [59, 60], Wang *et al* further focus on the student population and analyze the correlation between the smartphone sensing data and student's mental states and academic performance. In addition to smartphones, other wearable devices such as wrist bands (e.g., Fitbit, Apple Watch) are also actively used for sensing our biometrics (e.g., heart rates) and activities (e.g., footsteps). Existing research has further extended its use to identify smoking behaviors [45], or authenticate users [12], or capture our body sounds [49], or sense pedestrian walking [24]. Our work differs in that our system does not require users to wear any on-body devices.

Sensing at Infrastructure Level. Another sensing paradigm is to set up sensing devices in the environment or reuse ambient signals to track user behaviors. There are three types of methodologies:

1) camera-based methods [15, 27, 28, 53, 55, 56, 57, 61, 62, 67, 68] including Kinect, which leverage RGB cameras or depth cameras to capture high-resolution images and depth information to track user motion. With images containing millions of pixels, these methods commonly entail expensive computational overhead to process images. Furthermore, the raw images can be leaked to the adversary [52, 66] and bring privacy and security concerns. StarLight relies on low-level, low-resolution visual clues to track user motion. Thus it entails much lower computational overhead and better protects user privacy.

2) RF-based methods [7, 8, 26, 40, 48, 58, 64, 73], which reuses wireless signals in the radio frequency (RF) to sense our gestures. They leverage either the multi-path effect [7, 8], or Doppler effect of radio signals [48], or the fine-grained channel state information [64], or directional antennas [40] to classify pre-defined gestures. Recent works [58, 26] have further studied tracking users in the 3D space. However, they are limited to tracking a single body part. In comparison, StarLight is not limited to pre-defined gestures and enables full-body tracking in real time.

3) light-based methods [33, 35, 36, 41, 69, 70, 72], which leverage the infrared or visible light to sense users. However, the majority of existing work tracks the 2D location of either the user body [33, 35, 69, 70] or the user finger [41, 72]. While [36] enables the reconstruction of the user skeleton in the 3D space, it requires a large number of sensors, is vulnerable to furniture blockage, and assumes a static user. StarLight addresses these practical issues to reconstruct a mobile user's 3D postures in real time, further pushing light-based human sensing closer to practice.

9. CONCLUSION

We presented StarLight, a light-based sensing system that exploits LED panels on the ceiling and a few photodiodes in the environment to recover fine-grained human skeleton gestures on the fly (40 FPS). StarLight greatly advances the prior design [36] by significantly reducing the number of intrusive sensors, overcoming furniture blockage, and supporting user mobility. We built StarLight using 20 LED panels and 20 photodiodes in a 3.6 m × 4.8 m office. Our results demonstrated its efficacy in practice.

We conclude by discussing the limitations of our current study and plans for future work. *First*, our current experiments focus on a single user. When multiple users move around, their shadows intersect and depart, which introduces not only significant computational overhead but also a slower convergence. We plan to leverage the power of GPU and consider inferring user's body parameters to help associate shadows with individual users. *Second*, our current study is under a fixed LED panel setting. We will vary the number and layout (e.g., placing panels at ceiling edges) of LED panels. We will also include LEDs inside floor lamps and wall lamps. They emit light rays in more diverse directions, facilitating the movement detection. *Third*, upon environmental changes (e.g., furniture removal or addition), we currently need to re-calibrate the system. To lower the recalibration overhead, we will consider only updating the placement of affected sensors. *Finally*, we are interested in features other than the flashing rate to differentiate LEDs. Inspired by recent works [5, 22] that exploit RFID's physical features to separate RFIDs, we will examine whether an LED exhibits unique physical features (e.g., rise time). It will remove the need of modulating LED lights and further lower the deployment overhead.

10. ACKNOWLEDGMENTS

We sincerely thank our shepherd Marco Gruteser and the reviewers for their insightful comments. We thank Ronald Peterson for his helpful suggestions on fabricating PCB boards. We also thank Zhao Tian, Chuankai An, Weichen Wang and Linfeng Wang for their support on our study. This work is supported in part by the National Science Foundation under grants CNS-1421528 and CNS-1552924. 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] MaxLite Direct Lit LED Flat Panels. http://www.maxlite.com/item/instruction-manual?=MLFP22DS4035.

[2] PDA10A photodetector. https://www.thorlabs.com/thorproduct.cfm?partnumber=PDA10A.

[3] Philips OneSpace luminous ceiling. http://www.lighting.philips.com/content/B2B_LI/en_AA/products/onespace.html.

[4] Texas Instrument OPT101. http://www.ti.com/product/opt101?qgpn=opt101.

[5] ABARI, O., VASISHT, D., KATABI, D., AND CHANDRAKASAN, A. Caraoke: An E-Toll Transponder Network for Smart Cities. In *Proc. of SIGCOMM* (2015).

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

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

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

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

[10] CHENG, X., DU, D.-Z., WANG, L., AND XU, B. Relay sensor placement in wireless sensor networks. *Wireless Networks 14*, 3 (2008), 347–355.

[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] CORNELIUS, C., PETERSON, R., SKINNER, J., HALTER, R., AND KOTZ, D. A Wearable System That Knows Who Wears It. In *Proc. of MobiSys* (2014).

[13] DIMITROV, S., AND HAAS, H. *Principles of LED Light Communications: Towards Networked Li-Fi*. Cambridge University Press, 2015.

[14] DONG, Y., HOOVER, A., MUTH, E., AND SCISCO, J. A new method for measuring meal intake in humans via automated wrist motion tracking. *Applied Psychophysiology and Biofeedback 37*, 3 (2012), 205–215.

[15] FANELLO, S. R., KESKIN, C., IZADI, S., KOHLI, P., KIM, D., SWEENEY, D., CRIMINISI, A., SHOTTON, J., KANG, S. B., AND PAEK, T. Learning to Be a Depth Camera for Close-range Human Capture and Interaction. *ACM Trans. Graph. 33*, 4 (July 2014).

[16] FEIGE, U. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM) 45*, 4 (1998), 634–652.

[17] 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).

[18] GUPTA, S., MORRIS, D., PATEL, S., AND TAN, D. SoundWave: Using the Doppler Effect to Sense Gestures. In *Proc. of CHI* (2012).

[19] 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).

[20] HOCHBAUM, D. S. *Approximation algorithms for NP-hard problems*. PWS Publishing Co., 1996.

[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, P., ZHANG, P., AND GANESAN, D. Laissez-Faire: Fully Asymmetric Backscatter Communication. In *Proc. of SIGCOMM* (2015).

[23] 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).

[24] JAIN, S., BORGIATTINO, C., REN, Y., GRUTESER, M., CHEN, Y., AND CHIASSERINI, C. F. LookUp: Enabling Pedestrian Safety Services via Shoe Sensing. In *Proc. of MobiSys* (2015).

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

[26] JOSHI, K., BHARADIA, D., KOTARU, M., AND KATTI, S. Wideo: Fine-grained device-free motion tracing using rf backscatter. In *Proc. of NSDI* (2015).

[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, D., IZADI, S., DOSTAL, J., RHEMANN, C., KESKIN, C., ZACH, C., SHOTTON, J., LARGE, T., BATHICHE, S., NIESSNER, M., BUTLER, D. A., FANELLO, S., AND PRADEEP, V. RetroDepth: 3D Silhouette Sensing for High-precision Input on and Above Physical Surfaces. In *Proc. of CHI* (2014).

[29] KLASNJA, P. V., CONSOLVO, S., MCDONALD, D. W., LANDAY, J. A., AND PRATT, W. Using Mobile & Personal Sensing Technologies to Support Health Behavior Change in Everyday Life: Lessons Learned. In *Proc. of American Medical Informatics Association Annual Symposium* (2009).

[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] KRAUSE, A., AND GUESTRIN, C. Near-optimal observation selection using submodular functions. In *AAAI* (2007).

[32] 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.

[33] 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).

[34] LANE, N. D., LIN, M., MOHAMMOD, M., YANG, X., LU, H., CARDONE, G., ALI, S., DORYAB, A., BERKE, E., CAMPBELL, A. T., AND CHOUDHURY, T. BeWell: Sensing Sleep, Physical Activities and Social Interactions to Promote Wellbeing. *Mobile Networks and Applications* (2014).

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

[36] LI, T., AN, C., TIAN, Z., CAMPBELL, A. T., AND ZHOU, X. Human sensing using visible light communication. In *Proc. of MobiCom* (2015).

[37] LLOYD, S. P. Least squares quantization in pcm. *Information Theory, IEEE Transactions on 28*, 2 (1982), 129–137.

[38] Lu, H., Frauendorfer, D., Rabbi, M., Mast, M. S., Chittaranjan, G. T., Campbell, A. T., Gatica-Perez, D., and Choudhury, T. StressSense: Detecting Stress in Unconstrained Acoustic Environments Using Smartphones. In *Proc. of UbiComp* (2012).

[39] Lu, H., Pan, W., Lane, N. D., Choudhury, T., and Campbell, A. T. SoundSense: Scalable Sound Sensing for People-centric Applications on Mobile Phones. In *Proc. of MobiSys* (2009).

[40] Melgarejo, P., Zhang, X., Ramanathan, P., and Chu, D. Leveraging Directional Antenna Capabilities for Fine-grained Gesture Recognition. In *Proc. of UbiComp* (2014).

[41] Moeller, J., and Kerne, A. ZeroTouch: An Optical Multi-touch and Free-air Interaction Architecture. In *Proc. of CHI* (2012).

[42] 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.

[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] Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming* (1978).

[45] Parate, A., Chiu, M.-C., Chadowitz, C., Ganesan, D., and Kalogerakis, E. RisQ: Recognizing Smoking Gestures with Inertial Sensors on a Wristband. In *Proc. of MobiSys* (2014).

[46] Pearson, K. Mathematical contributions to the theory of evolution. iii. regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London. Series. A 187* (1896), 253–318.

[47] Post, E. R., Orth, M., Russo, P. R., and Gershenfeld, N. E-broidery: Design and fabrication of textile-based computing. *IBM Systems Journal* (2000).

[48] Pu, Q., Gupta, S., Gollakota, S., and Patel, S. Whole-home gesture recognition using wireless signals. In *Proc. of MobiCom* (2013).

[49] Rahman, T., Adams, A. T., Zhang, M., Cherry, E., Zhou, B., Peng, H., and Choudhury, T. BodyBeat: A Mobile System for Sensing Non-speech Body Sounds. In *Proc. of MobiSys* (2014).

[50] Rajagopal, N., Lazik, P., and Rowe, A. Visual light landmarks for mobile devices. In *Proc. of IPSN* (2014).

[51] Richman, E. E. Requirements for lighting levels, 2012.

[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* (2013).

[53] Sharp, T., Keskin, C., Robertson, D., Taylor, J., Shotton, J., Kim, D., Rhemann, C., Leichter, I., Vinnikov, A., Wei, Y., Freedman, D., Kohli, P., Krupka, E., Fitzgibbon, A., and Izadi, S. Accurate, Robust, and Flexible Real-time Hand Tracking. In *Proc. of CHI* (2015).

[54] Shotton, J., et al. Real-time human pose recognition in parts from single depth images. In *Proc. of CVPR* (2011).

[55] Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., and Moore, R. Real-time human pose recognition in parts from single depth images. *Communications of the ACM 56*, 1 (2013), 116–124.

[56] Song, J., Pece, F., Sörös, G., Koelle, M., and Hilliges, O. Joint Estimation of 3D Hand Position and Gestures from Monocular Video for Mobile Interaction. In *Proc. of CHI* (2015).

[57] Song, J., Sörös, G., Pece, F., Fanello, S. R., Izadi, S., Keskin, C., and Hilliges, O. In-air Gestures Around Unmodified Mobile Devices. In *Proc. of UIST* (2014).

[58] Sun, L., Sen, S., Koutsonikolas, D., and Kim, K.-H. WiDraw: Enabling Hands-free Drawing in the Air on Commodity WiFi Devices. In *Proc. of MobiCom* (2015).

[59] Wang, R., Chen, F., Chen, Z., Li, T., Harari, G., Tignor, S., Zhou, X., Ben-Zeev, D., and Campbell, A. T. StudentLife: Assessing Mental Health, Academic Performance and Behavioral Trends of College Students Using Smartphones. In *Proc. of UbiComp* (2014).

[60] Wang, R., Harari, G., Hao, P., Zhou, X., and Campbell, A. T. SmartGPA: How Smartphones Can Assess and Predict Academic Performance of College Students. In *Proc. of UbiComp* (2015).

[61] Wang, R., Paris, S., and Popović, J. 6D Hands: Markerless Hand-tracking for Computer Aided Design. In *Proc. of UIST* (2011).

[62] Wang, R. Y., and Popović, J. Real-time hand-tracking with a color glove. *ACM Transactions on Graphics* (2009).

[63] Wang, W., Liu, A. X., Shahzad, M., Ling, K., and Lu, S. Understanding and Modeling of WiFi Signal Based Human Activity Recognition. In *Proc. of MobiCom* (2015).

[64] Wang, W., Liu, A. X., Shahzad, M., Ling, K., and Lu, S. Understanding and modeling of wifi signal based human activity recognition. In *Proc. of MobiCom* (2015).

[65] 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).

[66] 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 *Security and Privacy (SP), 2011 IEEE Symposium on* (2011), IEEE, pp. 147–161.

[67] Wu, Y., and Huang, T. S. View-independent recognition of hand postures. In *Proc. of CVPR* (2000).

[68] Wu, Y., Lin, J. Y., and Huang, T. S. Capturing natural hand articulation. In *Proc. of ICCV* (2001).

[69] Xie, B., Tan, G., and He, T. Spinlight: A high accuracy and robust light positioning system for indoor applications. In *Proc. of SenSys* (2015).

[70] Yang, Z., Wang, Z., Zhang, J., Huang, C., and Zhang, Q. Wearables Can Afford: Light-weight Indoor Positioning with Visible Light. In *Proc. of MobiSys* (2015).

[71] Younis, M., and Akkaya, K. Strategies and techniques for node placement in wireless sensor networks: A survey. *Ad Hoc Networks 6*, 4 (2008), 621 – 655.

[72] Zhang, C., Tabor, J., Zhang, J., and Zhang, X. Extending Mobile Interaction Through Near-Field Visible Light Sensing. In *Proc. of MobiCom* (2015).

[73] Zhao, C., Chen, K.-Y., Aumi, M. T. I., Patel, S., and Reynolds, M. S. SideSwipe: Detecting In-air Gestures Around Mobile Devices Using Actual GSM Signal. In *Proc. of UIST* (2014).