Arrays of single pixel time-of-flight sensors for privacy preserving tracking and coarse pose estimation

Indrani Bhattacharya and Richard J. Radke
Department of Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute
bhatti@rpi.edu, rjradke@ecse.rpi.edu

Abstract

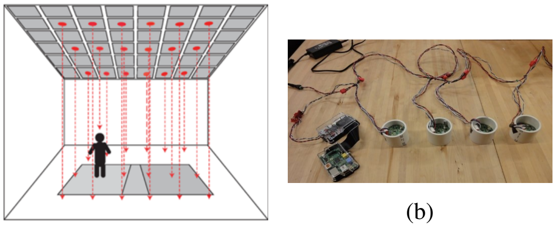
We present a method for real-time person tracking and coarse pose estimation in a smart room using a sparse array of single pixel time-of flight (ToF) sensors mounted in the ceiling of the room. The single pixel sensors are relatively inexpensive compared to commercial ToF cameras and are privacy preserving in that they only return the range to a small set of hit points. The tracking algorithm includes higher level logic about how people move and interact in a room and makes estimates about the locations of people even in the absence of direct measurements. A maximum likelihood classifier based on features extracted from the time series of ToF measurements is used for robust pose classification into sitting, standing and walking states. We use both computer simulation and real-world experiments to show that the algorithms are capable of robust person tracking and pose estimation even with a sensor spacing of 60 cm (i.e., 1 sensor per ceiling tile).

1. Introduction

Environments that feature "lighting systems that think" are becoming a reality thanks to a fusion of advanced light sources, sensors and integrated control systems. These smart lighting systems not only provide greater energy savings via the efficient use of lighting (which currently consumes 19% of electrical energy globally [17]), but also promote improved health, productivity and well-being. However, the benefits of a smart lighting system go beyond mere energy efficiency; the goal is to design smart rooms that can automatically detect human needs and react by providing the right light where and when it is needed. Smart rooms will be capable of automatically creating task-specific lighting by understanding different human usage patterns in a room (e.g., reading a book, working on a laptop, attending or presenting at a meeting), thereby serving the dual purposes of increased energy savings and increased productivity. Thus, robust, real-time room occupancy sensing is a critical aspect of smart lighting systems.

Guo et al. surveyed different occupancy sensing systems for lighting applications [8], highlighting the use of passive infrared (PIR) sensors, audible sound sensors, microwave sensors, and video cameras. It is critical to note that video cameras, while commonly used in computer vision applications, are generally not suitable for smart room applications in which the occupants have a reasonable expectation of privacy, and definitely not suitable for sensitive environments like nursing homes or restrooms. PIR sensors are the most common choice for occupancy detection, but are rarely deployed alone since they require motion to trigger. Hence, in most practical applications, they are used in conjunction with other sensors. For example, Agarwal [2] described a situation where PIR sensors are used in combination with magnetic reed switch door sensors to detect occupancy, which was used for controlling HVAC systems. Pandharipande and Caicedo [14] presented an ultrasound array sensor with enhanced presence detection capability to determine occupant locations. Neither PIR nor ultrasound sensors are well-suited to estimating finer-level details like the locations, poses and activities of people in a room, which would be required for optimal lighting control.

In this paper, we propose to use a sparse array of downward-pointed single pixel time-of-flight (ToF) sensors to estimate human occupancy and coarse pose in real time, without sensing any information that could compromise the occupants' privacy. Each sensor detects the height of any object under it, as illustrated in Figure 1. Our algorithms are based on simple image processing, pattern recognition, and logic about how people naturally move and interact. We test our algorithms in a simulated lab environment and also in a real, physical testbed, demonstrating that robust occupancy tracking and pose estimation are possible using even this sparse sensor arrangement.



(a)

(b)

Figure 1: (a) The concept of a smart ceiling panel with single pixel range sensors. The red spots denote the positions of the sensors and the dashed lines indicate time-of-flight rays. (b) Multiple LeddarOne sensors connected in series for physical deployment.

2. Related Work

In traditional computer vision, cameras and range sensors are widely used to capture images, videos, and depth maps of a scene. Time-of-flight (ToF) cameras that provide 3D measurements of a scene have been used for tracking occupants [7] and recognizing poses [6] and gestures in smart rooms. Booranrom et al. [3] developed a smart bedroom using Microsoft Kinect sensors to assist the elderly to turn on and off remote devices without touching them or using remote controls. The system also monitored signs of abnormality in the behavior of the elderly occupant. Fernandez et al. [5] presented an interactive room using Kinect sensors that could infer pointing direction from the user's elbow-wrist vector. Wu et al. [19] used machine learning to recognize different human activities in daily living environments based on inputs from Kinect sensors.

Despite significant recent research and applications of depth cameras such as the Kinect, these devices are generally not suitable for smart lighting applications due to their cost and the privacy issues involved; a high-resolution range image allows significant information to be observed about an occupant. Privacy preserving occupancy sensing for smart lighting applications was recently addressed by Wang et al. [18], who used modulated light and distributed color sensors to estimate the 3D occupancy of an indoor space. Dai et al. [4] presented a method for activity recognition in a smart room using extremely low resolution cameras. Most relevant to this paper, Jia and Radke [10] introduced the idea of downward-pointed ToF rays for occupancy and pose estimation. Privacy preservation was investigated by downsampling a full-resolution ToF camera to mimic a sparse array, and by simulating larger environments and sensor arrays using a game engine.

In contrast, in this paper, we designed an actual array of single-pixel ToF sensors, mounting one sensor per 60 cm×60 cm ceiling tile, in keeping with the vision of Figure 1a, for real-time person tracking and coarse pose estimation. The single-pixel sensors are a much cheaper alternative to the ToF cameras used in [10], and inherently preserve privacy since there is no direct sensing in large regions between the ToF rays. The difference between the real downward-pointed array and the downsampled fan-beam reconstruction of [10] also required the development of robust algorithms for height and pose estimation, as described in Section 4.

3. System setup

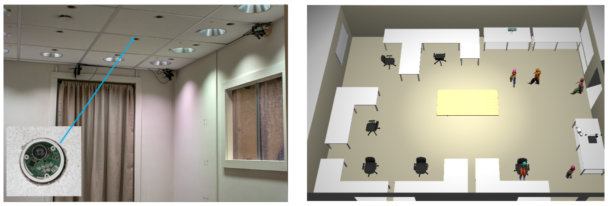
3.1. Real testbed

In our experiments, we used the LeddarOne Sensing module [11] manufactured by LeddarTech, illustrated in Figure 1b. The LED emitters emit infrared light (850 nm) pulsed at high frequency; the single channel receiver collects the backscatter of the emitted light and measures the time taken for the emitted light to return back to the sensor. The sensor has a 3° conic beam angle and is entirely dedicated to one measurement. The distance to objects falling within the beam can be measured by analyzing the first detection of the returning waveform. Outside of the smart lighting application discussed here, these single-pixel ToF sensors have also found application as an effective non-intrusive solution for continuously detecting the level of water in cooling towers [12] and proximity detection and distance measurement in overhead monorail conveyor systems [13].

We prepared mounts for the sensors, including a PVC pipe to enclose the sensors and an infrared transmitting acrylic sheet. We designed a wire harness for connecting 20 LeddarOne sensors and mounted them on the ceiling of our physical testbed, a 2.4 m × 3.7 m × 2.4 m experimental room used to develop and explore adaptive lighting strategies in a controlled environment. The sensors were connected to the wire harness in series and powered by a single 5V power supply. We used a single RS 485 to USB converter for data acquisition using the Modbus protocol at a baud rate of 115200. In this configuration, the average frame rate for polling data from all the sensors is approximately 2 fps. The LeddarOne sensors are mounted such that there is one sensor per ceiling tile in a 4×5 grid. The physical testbed with installed LeddarOne sensors is shown in Figure 2a.

3.2. Simulated testbed

In order to investigate the performance of our algorithm in larger environments, we also developed a 3D simulated office environment using the Unity game engine [16]. The 18 m × 14 m simulated space is shown in Figure 2b. Single pixel time-of-flight sensors are simulated as the center pixels of downward looking orthographic cameras spaced at 60 cm apart to mimic the effect of 1 sensor per ceiling tile.



(a) (b)

Figure 2: (a) The physical testbed with mounted sensors, (b) 3D simulated lab environment designed using the Unity game engine.

Downward pointing rays are cast from the simulated sensors and the range to hit points gives the distance measurement as seen by the sensors. The sensors were simulated based on the behavior of the actual LeddarOne sensors used in our real experiments.

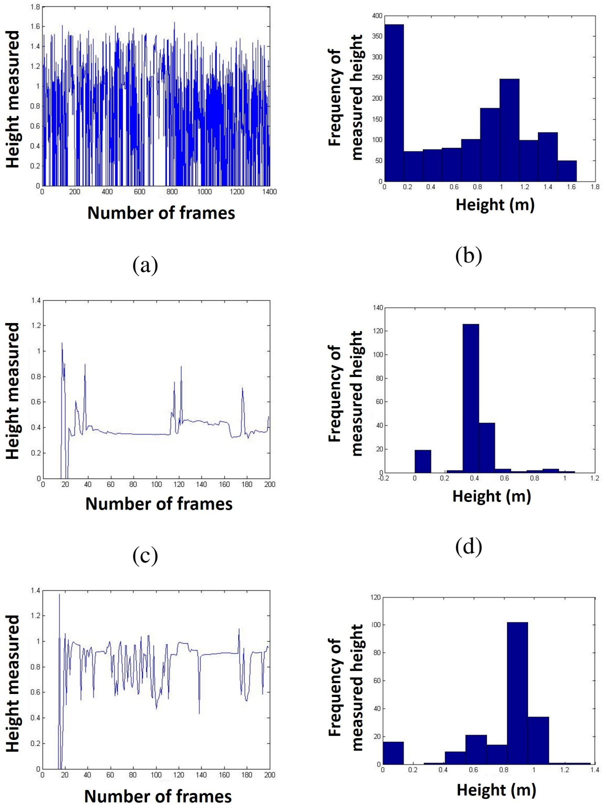
The simulated environment mimics the natural movement patterns of individuals in a typical office environment, with several people entering the room, walking around, sitting down, standing up, standing close to each other, gathering around the conference table and in front of the experimental bench, and leaving the room. It also allows us to specify the frame rate of data collection, so that the performance of the algorithms as a function of this parameter can be estimated.

4. Real time tracking and pose estimation

Our objective is to detect people, track their positions, and estimate their coarse poses in real time from the distance measurements obtained from the sensors. In a room of 2.5 m height, each sensor only covers a floor area of approximately 13 cm × 13 cm. Thus, when spaced at 60 cm apart (the average distance between typical commercial ceiling tiles), there are large “blind spots” in between the sensors where people get lost. We apply logic about how people actually move and interact in the room in such situations to estimate the position of the person in the blind spot.

We first recorded the sensor measurements for a walking person and a sitting person in the testbed to investigate how the measurements vary. Figure 3 shows the time series and the histogram of the height measurements of a person walking and sitting in the room. If we observe the time series pattern of the measurements, we note that significant information can be obtained about the state of the occupant.

Every time an occupant moves to a blind spot, there is no measurement, denoted as a zero. Figure 4a shows this situation when Person 1 is detected by the sensors, but Person 2, being out of the field of view of the sensors, is not detected. The likelihood of a missed reading is higher for a walking person. From the histogram of Figure 3b, we see that roughly 25% of the time, a walking person is in a blind spot and there is no height measurement for this person at all. Also, for a moving person, the variation in recorded measurements is significantly higher than for a stationary person. This is because the reflected ray to the sensor could be from any part of the body. From a histogram of the mea-

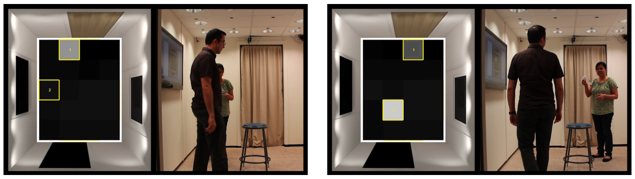


(a) (b)

(c) (d)

(e) (f)

Figure 3: (a) Time series and (b) histogram of height measurements obtained from the LeddarOne sensors for a walking person. (c) Time series and (d) histogram of height measurements while sitting on a low chair. (e) Time series and (f) histogram of height measurements while sitting on a high chair.



(a) (b)

Figure 4: Real time tracking in the testbed; (a) Person 1 is detected by the sensors, Person 2 is not detected by any sensor. (b) The shoulder of Person 1 is detected, while the head of Person 2 is detected.

surements, we observe that for a moving person, most of the time, the sensor detects either the head or the shoulder, corresponding to the 2 peaks seen in the histogram of Figure 3b. Figure 4b shows a situation where the sensors detect the head of Person 2 (denoted by the lighter rectangular blob) and the shoulder of Person 1 (denoted by the darker blob).

The height of a sitting person is largely dependent on where he or she is sitting, since single pixel sensors return a single measurement (or sometimes no measurement at all) per person. Figures 3c–d and e–f show the cases where the person is sitting on a low chair and a high chair respectively. It is evident from these graphs that at any instant, the measured height alone cannot give robust tracking and cannot estimate the pose of the occupant. Hence, we need to estimate the location and pose of the occupant based on previous measurements.

4.1. Pre-processing

The single pixel time-of-flight sensors return the range to hit points. These measured distances, when subtracted from the known height of the room, form an elevation map of the room. It is assumed that we have a background elevation map of the room that contains no people or movable furniture, denoted as *B(x,y)*, which can be acquired during system calibration. An elevation map of the foreground objects in the scene $F_t(x, y)$ can be obtained from the current elevation map $E_t(x, y)$ by background subtraction and thresholding:

$$F_t(x, y) = \begin{cases} 0, & \text{if } |E_t(x, y) - B(x, y)| \leq \tau \\ E_t(x, y), & \text{otherwise} \end{cases}$$

The threshold τ is set to 10 cm, which is lower than the height of a human, even if he or she is lying on the floor.

4.2. Real-time tracking

People can be easily detected in the thresholded foreground image. Assuming that each person walks into the room, each person entering the scene is detected at the image boundary corresponding to the door location and given a unique label. Each person blob is then tracked throughout the following images until it leaves the frame. The position of the person is estimated as the location of the sensor that sees him. Every person blob in the current frame is compared with the person blobs in the previous frame to find the closest spatial match using the Hungarian algorithm. Labels are propagated from one frame to the next. It may happen that in the current frame, the number of detected people in the room is less than that in the previous frame. This typically happens when someone moves into a blind spot and none of the sensors can see him. Since people cannot spontaneously appear or disappear from the room, in the case of no readings corresponding to a person, his location at the missing data frame is assumed to be the same as that of the last frame. From our experiments, we found that this assumption is reasonable, since a person can only move within a specified radius while moving at a constant velocity (typically his regular walking pace) in between successive frames. The estimated position is quickly updated as soon as real measurements are received. As we show in Section 5, this relatively simple algorithm is quite accurate for our motivating application of lighting control; more complex and computationally demanding Kalman or particle filters for tracking are not necessary. The algorithm for real-time tracking and pose classification is given in Algorithm 1.

Algorithm 1: Real time tracking and pose estimation in a smart room

```
Input  : Foreground elevation map $F_t(x, y)$
Output: Real time location and pose estimate at time t
for $t \leftarrow 1, 2, \ldots$ do
    Extract the blobs, record the location and heights
    if the extracted blob appears near door and is
      outside a specified radius of any of the blobs from
      frame $t - 1$ then
        Assign new label to the blob
    end
    Match the position of every extracted blob in
        frame $t$ with blobs in the frame $t - 1$ using
        the Hungarian algorithm
    Pass the labels of the blobs in frame $t - 1$ to
        the blobs in frame $t$
    if blob in frame $t - 1$ cannot be matched to any of
      the blobs in frame $t$ then
        Estimate a position for the blob in frame $t$
        based on the detected position in frame $t - 1$
    end
    if blob is labeled then
        Classify poses using Algorithm 2
    end
end
```

4.3. Pose estimation

Our goal in this section is to use the time series of the measurements as the basis for classifying each tracked object into standing/walking and sitting categories. At each instant t, we keep a record of the last n measurements (in our experiments, we choose $n = 40$, as discussed in Section 5.1). Let this vector of n measurements be denoted as $X(t)$. A median filter of size 5 frames is applied to $X(t)$; we denote the median filtered data as $X'(t)$. The median filter removes short duration missing readings from the set of measurements. Based on our observations from Figure 3, we find that the mean or the median of $X(t)$ or $X'(t)$ is not sufficient as a feature alone in distinguishing the two categories. This is illustrated by Figures 5a and 5b, where we present the measurements for a particular window. We note that the mean and median of the filtered height measurements for a walking person (0.51 m and 0.62 m respectively) are lower than those of a person sitting on a high chair (0.92 m and 0.91 m respectively), while in reality a standing/walking person should have a height greater than when he is sitting. The mean and median for a person sitting on a low chair are 0.44 m and 0.45 m respectively, as given in Figure 5c. We note that the likelihood of missing readings and spikes in measurements is higher for the moving person, and hence conclude that the variance of the obtained measurements is a good feature for distinguishing between the stationary and the moving person. On the other hand, the variance alone cannot distinguish between a person standing still and a person sitting on a chair. Since using either of the two features alone cannot solve our pose classification problem, we thus consider the feature vector $\mathbf{x} = [x_1, x_2]$, where $x_1 = var(X(t))$ and $x_2 = mean(X'(t))$ for our pose classifier.

Let ω_1 and ω_2 denote the states standing/walking and sitting respectively. Let $p(\mathbf{x}|\omega_i)$ be the state conditional probability density function for **x**. We apply a simple maximum likelihood classifier to estimate the coarse pose of each person:

$$Decide \begin{cases} \omega_1, & \text{if } p(\mathbf{x}|\omega_1) > p(\mathbf{x}|\omega_2) \\ \omega_2, & \text{otherwise} \end{cases}$$

We model the class-conditional densities $p(\mathbf{x}|\omega_i)$ as bivariate Gaussian distributions, whose parameters were estimated from 1000 frames of simulated training data and 1000 frames of real training data from each class. We also assumed that the off-diagonal elements in the covariance matrix are zero, i.e., the features are independent.

Due to people moving in the blind regions every now and then, the classifier might change the pose label as soon as there is a missed reading or noise in the measurement. In order to change the label only when there is compelling evidence for a transition of state, we add latency to the system. If the changed label sustains for c frames, we are convinced that the transition of state has occurred in reality and the classification is not due to sporadic missed readings or noise. We discuss the choice of c in Section 5.1.

The pseudocode for the pose classification is given in Algorithm 2. In the next section, we present the results for the real-time tracking and pose classification algorithms.

5. Experimental Results

In order to test our tracking and pose classification algorithms, we collected several simulated and real datasets for our experiments. These datasets are entirely separate from those we used for training and parameter tuning.

Algorithm 2: Pose classification of a person in a smart room at time t

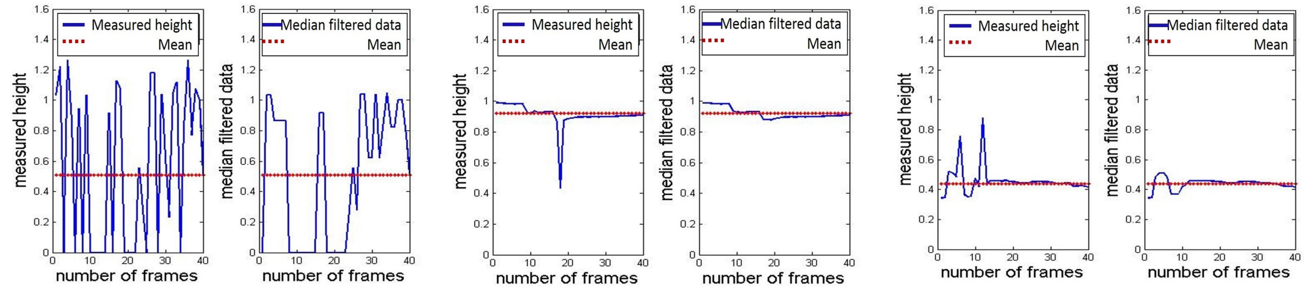
```
Input  : $X(t)$ , $p(\mathbf{x}|\omega_i)$
Output: Pose estimate at time t
Median-filter $X(t)$ to obtain $X'(t)$
Compute $x_1 = var(X(t))$, $x_2 = mean(X'(t))$
Form the feature vector $\mathbf{x} = [x_1, x_2]$
Classify poses using maximum likelihood classifier
    as given in Section 4.3
if $pose(t) \neq pose(t-1)$ then
    Wait and observe pose for next $c$ frames
    for compelling evidence of transition of state
end
if new state persists for c frames then
    Change pose to that found by the maximum
    likelihood classifier
end
else
    Retain same pose as previous frame,
    $i.e., pose(t) = pose(t-1)$
end
```

Dataset 1 : Simulated video, 6000 frames, 1 person, frame rate of sensors: 10 fps.
Dataset 2 : 4 simulated videos, 3000 frames each, 1 person, frame rate of sensors: 1, 5, 15, 30 fps.
Dataset 3 : Simulated video, 1500 frames, 5 people, frame rate of sensors: 10 fps.
Dataset 4 : Real video, 3000 frames, 2 people, frame rate of sensors: 2 fps.

5.1. Experiments on Dataset 1

We recorded 6000 frames of video of the simulated lab, having the sensors poll data at 10 frames per second, with a single person walking in, sitting down, moving around, leaving and re-entering the room. We recorded the elevation maps that mimic the output from the real time-of-flight sensors directly from the game engine.

We studied the effect of changing the latency c and the window size n (i.e., the number of previous frames which we consider for constructing the feature vector) on the pose classification performance. As n is decreased with $c = 30$ frames, the number of false alarms increases, even though the 6 actual pose transitions are correctly detected. Again, when c is decreased keeping $n = 40$ frames, the number of false alarms increases. The results are shown in Figure 6. Thus, we conclude that at a sensor data rate of approximately 10 fps, we need the window size and the latency to be only a few seconds for accurate pose classification. Since lighting applications should only respond after a pose transition has been confirmed and persists for some time, this



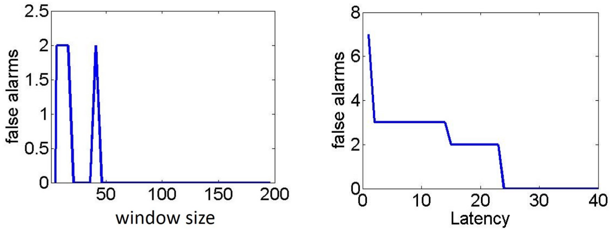
(a)

(b)

(c)

Figure 5: (a) Measured heights and median filtered measurements for a walking person, (b) Measured heights and median filtered measurements for a person sitting on a high chair, (c) Measured heights and median filtered measurements for a person sitting on a low chair.

latency is quite acceptable. For all our future experiments, we used a window size of $n = 40$ frames and a latency in decision $c = 30$ frames.



(a)

(b)

Figure 6: (a) Effect of changing window size n keeping latency $c = 30$. (b) Effect of changing c keeping $n = 40$.

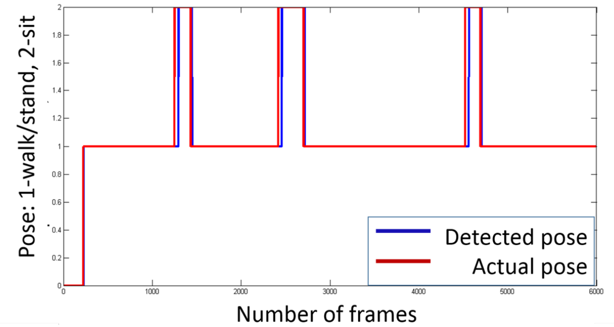


Figure 7: Detected/actual pose transitions for Dataset 1.

With $n = 40$ frames and $c = 30$ frames, we observe that the algorithm correctly tracks the person and classifies the poses even when he is lost between the sensors. The algorithm could detect all entrances and exits and all pose transitions, with no missed detections or false alarms. The errors in location are largely due to estimates made in the blind spots. The mean error in estimated location, calculated as the absolute difference between the estimated position and the exact position given by Unity, is 0.42 m, which is less than the sensor spacing of 0.6 m. The median error in location is 0.35 m. The mean error is more than the median error because of the short durations of missing readings when the error in location shoots up. Figure 7 shows the ground truth and estimated poses; these are in agreement for 97% of the frames, and the disagreements are due to the delays in estimation. The average delay in detecting an actual transition for standing to sitting states is 44 frames (4.4 sec) and in detecting an actual transition from sitting to standing states is 16 frames (1.6 sec).

5.2. Experiments on Dataset 2

In this experiment, we studied the effect of varying the frame rate of the sensors in the simulated environment on the performance of our algorithm. The video contains a single person walking into a room and sitting down on a chair. The results are given in Table 1.

As expected, higher frame rates lead to better localization and less delay in detecting transitions. However, even at low frame rates, the error and delay would be quite acceptable for our motivating application of lighting control.

Table 1: Performance for Dataset 2.

Frame rate of the sensors	1	5	15	30
Mean error (m)	0.59	0.48	0.45	0.38
Detection delay (sec)	42	10	3.5	1.8

5.3. Experiments on Dataset 3

We next recorded 1500 frames of simulated video with sensors polling data at 10 fps where 5 people walk in, sit down, come close together, separate and leave the room. Figure 8 shows snapshots of the simulated scene and tracking results. Black blobs indicate that the person is lost between the sensors but his/her position is estimated. Table

2 shows the performance comparison with respect to the ground truth. At one point, Person 3 sits down in a chair located in a blind spot and keeps sitting for some time as depicted in Figure 8. Since there are no measurements for this person for quite some time and we know that he has not left the scene, his position is estimated at his last seen position and his pose is the last detected pose (i.e., standing). This is why the algorithm fails to detect the sitting down and standing up of Person 3.

Table 2: Performance for Dataset 3.

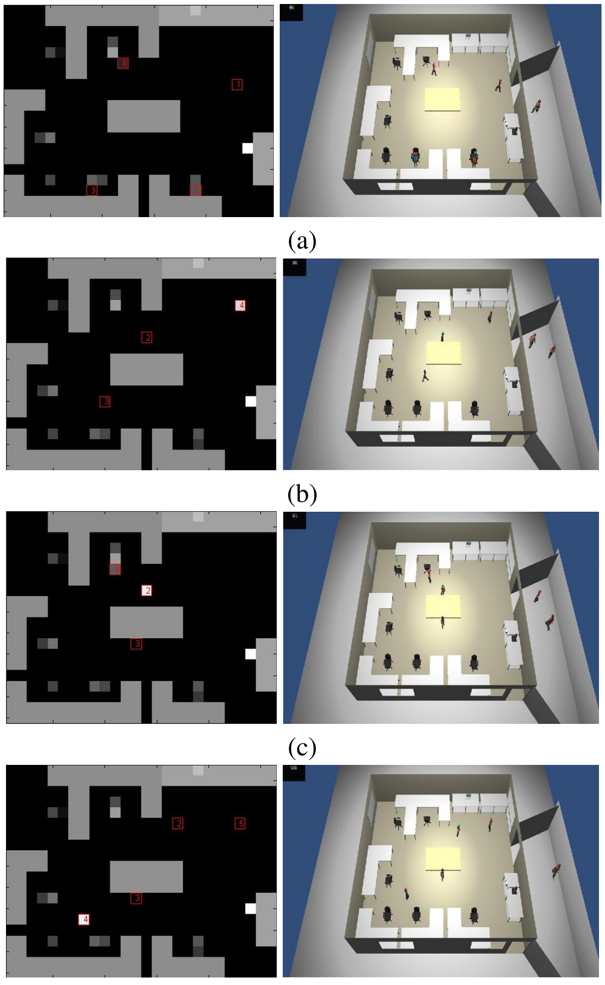
	Ground Truth	Detected	False Alarm
Entrance	5	5	0
Exit	1	1	0
Stand up	2	1	0
Sit down	2	1	0

5.4. Experiments on Datasets 4

We tested our algorithm for real-time tracking and pose estimation in the real physical testbed described in Section 3.1. We used the distance measurements obtained from the 20 LeddarOne sensors in the ceiling of the physical testbed (one sensor per ceiling tile) for our algorithms. The frame rate varied between 1-2 fps. Several snapshots of real time tracking are shown in Figure 9. The tracking results are shown on the left with the actual scene on the right.

Since the frame rate is between 1-2 fps, the probability of a person being lost between the sensors is higher than in the simulated environment. Even so, the mean error in location is around 0.5 m, which is very close to our results from the simulated experiments at the same sensor frame rate. Also, since the room is small, labels are more frequently interchanged as more people enter the room and walk close to each other. However, for lighting control applications, we are not so much concerned with maintaining the identity of the person blobs as we are with accurately detecting their position and pose. The pose classification algorithm is able to detect all transitions correctly; however, in this case, the latency in decision at 1 fps becomes approximately 40 seconds, i.e., the algorithm waits for 40 seconds before actually transitioning from one pose to another. This latency is expected to suffice for lighting control applications. [1]

While our work is closely related to the work of Jia and Radke [10], it is difficult to directly compare the performance of our real testbed with [10] due to inherent differences in the system setups. While they studied privacy preservation by spatially downsampling a 176x144 image from a single ToF camera, we actually built a sparse array of single-pixel, small-cone-angle sensors. The differing fan-beam and parallel-beam geometries result in qualitatively different distance readings. The floor coverages of the setups are also different, being more limited in [10]. Also, the narrow-FOV SwissRanger SR4000 ToF camera used in [10] cost $4295, while our LeddarOne sensors cost $175 each.


(a)

(b)

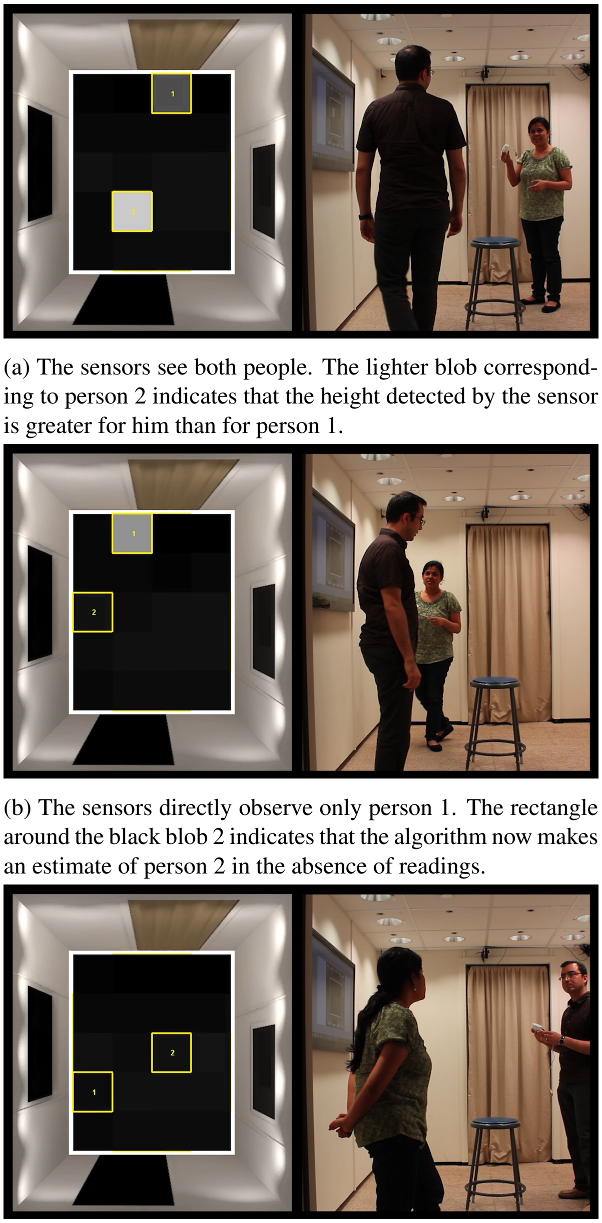
(c)

(d)

Figure 8: Snapshots of tracking in Dataset 3. The right side diagrams show snapshots from Unity and the left side diagrams show the tracking result in the corresponding frames. (a) 4 people detected in the room with person 2 and person 3 seated. Person 3 sits in a blind spot and hence is not picked up by any of the sensors, as indicated by the black blob. (b) 3 people detected in the room; person 1 has left the room. (c) 3 people in the room; sensor sees only 2; position of the 3rd person is estimated. (d) 4 people in the room, a new person 5 enters the room. Only 1 person is seen by the sensor; the positions of the other 3 are estimated.

However, our algorithm can be directly compared with [10] in terms of simulated experiments. In [10], the sensor

[1] Short video clips demonstrating tracking in the SST and the simulated lab can be found at `https://goo.gl/603afj`.



(a) The sensors see both people. The lighter blob corresponding to person 2 indicates that the height detected by the sensor is greater for him than for person 1.

(b) The sensors directly observe only person 1. The rectangle around the black blob 2 indicates that the algorithm now makes an estimate of person 2 in the absence of readings.

(c) The sensors do not see either of the people but the algorithm makes an estimate of both of their positions based on previous readings.

Figure 9: Real time tracking in the physical testbed.

spacing was varied from 1 to 40 cm, at a data rate of 30 fps. In contrast, we studied the performance of our system at a sensor spacing of 60 cm, similar to a single sensor per ceiling tile, and varied the data acquisition rate from 1 to 30 fps. At 30 fps, the location accuracies for both systems are comparable; 0.2 m accuracy for a sensor spacing of 40 cm in [10] against 0.38 m accuracy for 60 cm sensor spacing in our system. In [10], the error rate for pose classification shoots up to 20% at a sensor spacing of 40 cm. Our algorithm classifies poses correctly even at 60 cm spacing after incorporating a delay (1.8 s at 30 fps, sufficient for lighting control), since it is based on careful analysis of observed data.

6. Conclusions and Future Work

We described a privacy-preserving method for accurate person tracking and coarse pose recognition using a sparse array of ceiling mounted single pixel ToF sensors. With proximity sensors in mobile phones beginning to use the time-of-flight principle [15] and miniaturized medium-range ToF cameras entering the market [9], single pixel ToF sensors are expected to become sufficiently cheap and low power to be integrated into commercial light fixtures.

A problem we faced with our real physical testbed is the low frame rate of the sensors (about 1-2 fps). The low frame rate leads to reduced accuracy and higher latency. We intend to work on increasing the frame rate of the sensors for greater accuracy. While we did not take advantage of it in this paper, the full-waveform analysis performed by the LeddarOne sensors provides the capability to detect multiple objects in the same region. This is possible if the closer object is smaller than the illuminated area for that region. The beam can then illuminate another object that is not completely "shadowed" by the closer object. In future work, we could leverage this information for more accurate height measurements.

We intend to investigate the performance of our algorithms in larger and more realistic environments. One possible area of investigation in this regard is the simulation of larger physical models (e.g., an entire office building) and more realistic human behavior using the Unity game engine, which can assist in designing task-specific lighting.

Ultimately, our vision of a smart lighting system involves a single troffer that will not only have color tunable LEDs for high quality light rendering, but also integrated multispectral sensors matched to the LEDs to enable advanced, distributed control, high speed modulated visible light for communication, and occupancy sensing using visible-light time-of-flight. As a first step towards this fusion, we plan to integrate the occupancy and pose detection framework described here with actual lighting control systems in a larger physical testbed [1].

7. Acknowledgments

This work was supported by the NSF under cooperative agreement EEC-0812056 and by New York State under NYSTAR contract C090145. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] S. Afshari, T.-K. Woodstock, M.H.T. Imam, S. Mishra, A.C. Sanderson, and R.J. Radke. The Smart Conference Room: An integrated system testbed for efficient, occupancy-aware lighting control. In *ACM International Conference on Embedded Systems For Energy-Efficient Built Environments (BuildSys)*, 2015.

[2] Y. Agarwal, B. Balaji, R. Gupta, J. Lyles, M. Wei, and T. Weng. Occupancy-driven energy management for smart building automation. In *2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, 2010.

[3] Y. Booranrom, B. Watanapa, and P. Mongkolnam. Smart bedroom for elderly using Kinect. In *ICSEC*, 2014.

[4] J. Dai, J. Wu, B. Saghafi, J. Konrad, and P. Ishwar. Towards privacy-preserving activity recognition using extremely low temporal and spatial resolution cameras. In *IEEE Workshop on Analysis and Modeling of Face and Gestures*, 2015.

[5] A. Fernandez, L. Bergesio, A. M. Bernardos, J. A. Besada, and J. R. Casar. A Kinect-based system to enable interaction by pointing in smart spaces. In *SAS*, 2015.

[6] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real time motion capture using a single time-of-flight camera. In *CVPR*, 2010.

[7] S. A. Gudhmundsson, M. Pardas, J. R. Casas, J. R. Sveinsson, H. Aanaes, and R. Larsen. Improved 3D reconstruction in smart-room environments using ToF imaging. *Computer Vision and Image Understanding*, 114(12):1376–1384, 2010.

[8] X. Guo, D. Tiller, G. Henze, and C. Waters. The performance of occupancy-based lighting control systems: A review. *Lighting Research and Technology*, 42(4):415–431, 2010.

[9] Heptagon Advanced Micro-optics. Micro-technologies for 3-d depth sensing systems. http://www.electronics-eetimes.com/en/depth-sensing-image-sensor-array-touted//-for-smartphones.html?cmp_id=7&news_id=222920044, 2015. [Online; accessed 02-September-2015].

[10] L. Jia and R. J. Radke. Using time-of-flight measurements for privacy-preserving tracking in a smart room. *IEEE Transactions on Industrial Informatics*, 10(1):689–696, 2014.

[11] LeddarTech. Leddarone, single element sensor module. http://leddartech.com/en/products-sensors/leddar-one-module. [Online; accessed 23-July-2015].

[12] LeddarTech. Detection and ranging for level-sensing applications. http://leddartech.com/files/documents/3d/e5/detection-and-ranging-for-level//-sensing-applications.pdf, 2014. [Online; accessed 21-July-2015].

[13] LeddarTech. Proximity detection and distance measurement for overhead monorail conveyor systems. http://leddartech.com/files/documents/ep/80/proximity-detection-and-distance//-measuremen-for-overhead-monorail//-conveyor-systems.pdf, 2014. [Online; accessed 21-July-2015].

[14] A. Pandharipande and D. Caicedo. Daylight integrated illumination control of LED systems based on enhanced presence sensing. *Energy and Buildings*, 43(4):944–950, 2011.

[15] STMicroelectronics. STMicroelectronics proximity sensor solves smartphone hang-ups. http://www.st.com/web/catalog/mmc/FM132/SC626/PF260441?icmp=pf260441_pron_p3609p_sep2014&sc=proximitysensor, 2013. [Online; accessed 21-July-2015].

[16] Unity. Unity gaming engine. https://unity3d.com/. [Online; accessed 23-July-2015].

[17] P. Waide, S. Tanishima, et al. *Light's Labour's Lost: Policies for Energy-efficient Lighting*. OECD Publishing, 2006.

[18] Q. Wang, X. Zhang, and K. L. Boyer. 3d scene estimation with perturbation-modulated light and distributed sensors. In *10th IEEE Workshop on Perception Beyond the Visible Spectrum*, 2014.

[19] H. Wu, W. Pan, X. Xiong, and S. Xu. Human activity recognition based on the combined SVM & HMM. In *ICIA*, 2014.