Using Time-of-Flight Measurements for Privacy-Preserving Tracking in a Smart Room

Li Jia, *Student Member, IEEE* and Richard J. Radke, *Senior Member, IEEE*

***Abstract*— We present a method for real-time person tracking and coarse pose recognition in a smart room using time-of-flight measurements. The time-of-flight images are severely downsampled to preserve the privacy of the occupants and simulate future applications that use single-pixel sensors in "smart" ceiling panels. The tracking algorithms use grayscale morphological image reconstruction to avoid false detections, and are designed not to mistakenly detect pieces of furniture as people. A maximum likelihood estimation method using a simple Markov model was implemented for robust pose classification. We show that the algorithms work effectively even when the sensors are spaced apart by 25cm, using both real-world experiments and environmental simulation.**

***Index Terms*— time-of-flight, smart room, pose recognition, visual tracking, privacy preservation, occupancy detection**

I. Introduction

Low-cost solid-state lighting technologies, computer vision algorithms, and advanced control systems are converging to make "smart rooms" — environments that react intelligently to the presence and activities of their occupants — a reality. In particular, we are concerned with reducing the cost and improving the efficient use of lighting, which currently consumes 19% of electrical energy globally [1]. About 20%-25% of the electricity used in buildings and about 5% of the total energy consumption in the US is used for lighting [2]. Energy-conserving lighting control systems for buildings and offices could cut these expenses to a large degree. For example, smart rooms could automatically create task-specific lighting (e.g., focusing light on the desk in front of a seated person and dimming the lights behind him/her), create lighting paths and cues for human assistance, or even detect anomalies in occupant behavior (e.g., alerting staff if an occupant in a nursing home suddenly falls over). While there have been significant advances in the computer vision research community in detecting human presence and classifying human activity from video, occupants would no doubt be unsettled by the impression that their room was "watching them". Therefore, we seek technologies that strike a balance between accurately detecting human presence, location, and pose, and preserving the privacy of a room's occupants.

Manuscript received July 26, 2012; revised November 25, 2012; accepted February 21, 2013. This work was supported in part by the National Science Foundation Smart Lighting Engineering Research Center, under the award EEC-0812056 and by New York State under NYSTAR contract C090145.

L. Jia and R.J. Radke are with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY, 12180. Email: jial@rpi.edu, rjradke@ecse.rpi.edu

In this paper, we propose to use ceiling-mounted, downward-pointed time-of-flight (ToF) sensors [3] to estimate human occupancy and pose in real time. The idea is to mount a sparse array of single-pixel range sensors on the ceiling of a room, which can detect the height of any object under them, as illustrated in Figure 1. We apply algorithms based on morphological image processing to this data for tracking humans, disambiguating humans from furniture, and coarse pose estimation. We also analyze how the algorithms' performance degrades as a function of element spacing and person density. The issue of element spacing is particularly important from the perspective of preserving privacy; we show that our algorithms produce accurate results even when the ToF elements are spaced at approximately 4×4 elements per square meter. Our ultimate goal is to use the prototype system described here to guide the design and deployment of "smart" ceiling panels, each of which contains a single ray of ToF depth information, as illustrated in Figure 1. We also discuss the use of realistic computer simulations to prototype detection and control algorithms, which can be used to investigate larger-scale deployments until smart panels are widely available.

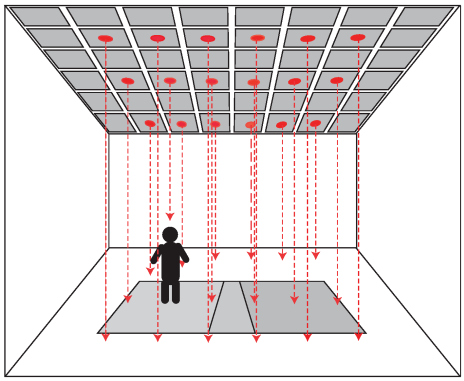

Fig. 1. Future smart ceiling panels, each mounted with a single-pixel depth sensor. The red spots represent the position where the ToF depth sensors are mounted, and the dashed lines indicate ToF rays.

II. Related Work

Real-time time-of-flight imaging is a topic of recent interest in the computer vision community, largely due to the increased availability of moderate-cost sensors. Time-of-flight approaches have the advantage of quick, straightforward information processing [3], and offer reasonably accurate low-resolution depth images at video frame rates, which can be quite useful for surveillance and human-machine interaction applications. On the other hand, compared to direct depth

sensing technologies such as LiDAR, time-of-flight cameras have less accuracy, may work poorly outdoors or at very close range, and typically exhibit a systematic distance error [4]. Computer vision researchers have recently made progress in applying super-resolution techniques to time-of-flight cameras, combining multiple successive images from a moving sensor to improve the quality of each frame [5].

Applications for smart rooms include tracking occupancy and recognizing poses and gestures. The first step for such applications is segmenting people from the background. Bianchi et al. [6] exploited the intensity signal produced by a ToF camera for foreground segmentation based on smart-seeded region growing and Kalman tracking, which allows for using a moving camera and multiple objects. Guðmundsson et al. [7] addressed the issue of real-time 3D reconstruction in a smart room, which creates more robust inputs for person tracking. They used a probabilistic background model based on ToF data to help in foreground person segmentation.

Real-time tracking and pose recognition in time-of-flight video has been addressed by several researchers. Gokturk and Tomasi [8] incorporated ToF depth data into a head tracking algorithm, reporting greater than 90% success rates. Malassiotis and Strintzis [9] described real-time 3D head pose estimation based on feature localization and tracking techniques using range data, which proved robust to illumination conditions. Hansen et al. [10] proposed a method for cluster tracking using a ToF camera in a smart environment, in which intensity and depth images are fused to build a background model. While this model doesn't distinguish different object types, the detection accuracy is high. Cai et al. [11] presented a regularized maximum likelihood deformable model that fitted color and depth images for 3D face tracking. Ganapathi et al. [12] derived a filtering method for human pose tracking using a side-looking ToF camera. A set of discriminatively trained patch classifiers is applied to detect body parts.

Despite this type of research into time-of-flight imaging, it's important to note that such systems are not in themselves suitable for smart lighting systems. The first problem is the limited field of view. A typical ToF camera has a field of view of $43^\circ \times 34^\circ$ [13]. Mounting the camera on the ceiling (2.5m high for a typical room) gives a coverage footprint of 2.0m$\times$1.6m on the floor. In order to cover a single office environment (e.g., 20m$\times$15m), we would need to mount over 100 cameras, which would be extremely expensive. Furthermore, real-time processing of the large amount of data generated by such an array would be computationally costly (and hence energy-consumptive). Finally, high-resolution coverage could lead to justifiable concerns in environments where occupants reasonably expect privacy. For these reasons, this paper focuses on methods that downsample the ToF sensor output as much as possible while maintaining reliable performance on localization and pose estimation problems relevant to smart lighting.

III. System Setup

In our experiments, we used a ceiling-mounted, downward-pointed SwissRanger SR4000 ToF camera produced by Mesa Imaging [14] as a proxy for the type of system illustrated in Figure 1. The camera emits RF-modulated (30MHz) near-infrared light (850nm), which is back-scattered by the scene and received by a CMOS CCD. The sensor continuously collects range and intensity measurements at a resolution of 176$\times$144 pixels. The angular resolution for each pixel is 0.39° at the center of the image, which corresponds to a 1.7 cm square on the ground, a 9 mm square on the head of a sitting person, and a 4.5 mm square on the head of a standing person (assuming a 2.5m ceiling). Since the ToF sensor emits divergent rays from a single center of projection, we modify each distance measurement by multiplying the raw sensor value by the cosine of the viewing angle to mimic orthographic projection. Distance measurements discussed in this paper always refer to this orthographic projection. The absolute accuracy in the distance measurements is approximately 10 mm within a range of 0.8–5.0 m.

The frame rate (FR) is related to the integration time (IT) and read-out time (RO) of the sensor by

$$FR = \frac{1}{4 \cdot (IT + RO)} \tag{1}$$

The integration time of the sensor plays a critical role; it should be set just high enough to maximize signal amplitude across the field of view without saturating the sensor. The signal amplitude depends on object distance and reflectance; the noise level at low-reflectance objects is relatively high. Another parameter, the amplitude threshold, defines the minimum amplitude that needs to be exceeded to accept a measurement. This is used to suppress noisy values where the pixels have low amplitude (reflectance), as well as around the edges of the sensor [15]. In our experiments, we used integration times of 6.3–8.3ms leading to frame rates of 23.4–19.4fps respectively. We set the amplitude threshold in the range 70–100 (on a scale of 0 to 16382).

We mounted the system in an office setting in which the height of the ceiling is 2.5m; the sensor's field of view is about 2.0m$\times$1.6m on the ground. Figure 2 illustrates an example image of the office environment and the corresponding time-of-flight image, which shows the measured distance from the sensor to each point in the scene.

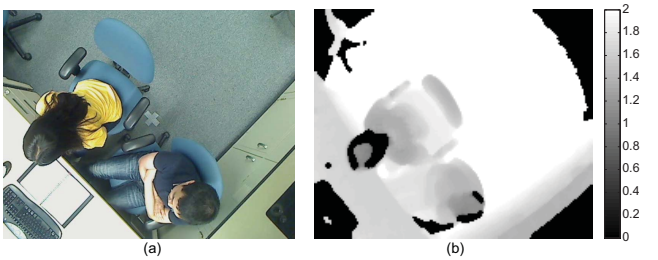

Fig. 2. (a) The office environment; the sensor is mounted in the ceiling above the desk. (b) The corresponding time-of-flight image, indicating distance from the sensor. The color bar represents measurements in meters. Entirely black pixels indicate missing measurements.

To investigate time-of-flight imaging in larger environments, we also created a 3D simulated lab space using the Unity

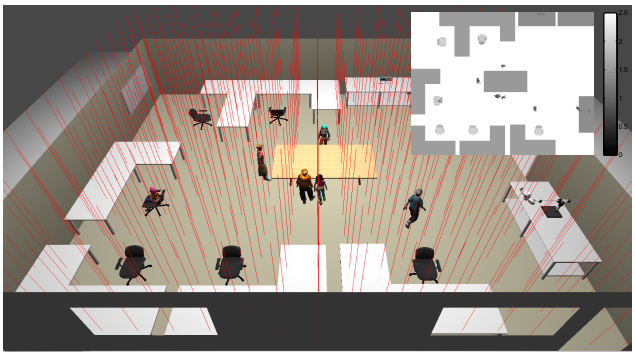

Fig. 3. The simulated environment of a lab with 5 people. The red lines denote the time-of-flight rays cast from the ceiling. The distance map (upper right corner) is created from sensors spaced 10cm apart.

game engine [16]. The 18×14 m^2 simulated space is illustrated in Figure 3. The distance images are created by casting rays downwards from simulated sensors that are spaced regularly on the ceiling and return the range to hit points. These images are processed by the algorithms we describe next in exactly the same way as the images output from the real ToF sensor.

We recorded several videos, both in the real and simulated environments, containing multiple people and pieces of furniture to design and test our algorithms. The most complicated real video contains four people entering the scene at different times, introducing and moving office chairs, sitting down close together, and leaving at different times. The simulated environment includes up to 30 people entering the lab, 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.

IV. Real-Time Tracking and Pose Estimation

Our main problems of interest are the real-time detection of humans, tracking of their positions, and estimation of their pose (i.e., either standing or sitting) from the distance measurements described above. Our approach combines morphological image processing algorithms with higher-level logic about how objects can move and interact. Each person entering the scene is detected at the image boundary and given a unique label. Each person blob is then tracked throughout the following images until it leaves the frame. Each person blob's location and height is measured in each frame, and its coarse pose is estimated from the height information. We begin by describing our overall approach to the problem on the full-resolution real-world video, and then analyze the algorithms' performance as the distance image is downsampled (simulating widely-spaced ToF elements) and as we move to the larger-environment simulation.

A. Pre-processing

While the integration time and amplitude threshold settings mentioned in Section III noticeably reduce the noise in the distance images, missing returns in low-reflectivity areas are still present (e.g., the person's dark hair and the keyboard on the desk in Figure 2a). We use a morphological "flood-fill" algorithm [17] to interpolate the lost values in any holes that are completely surrounded by known pixels, though some regions of missing data that are not totally surrounded remain.

We obtain a real-time elevation map by subtracting the real-time distance images from the known height of the ceiling. We also assume that we have a background elevation map that contains no people or movable furniture, denoted as $B(x, y)$, which can be acquired during system calibration. From the current elevation map $E_t(x, y)$, we obtain an elevation map of the foreground objects in the scene $F_t(x, y)$ using background subtraction and thresholding:

$$F_t(x,y) = \begin{cases} 0 & \text{if } |E_t(x,y) - B(x,y)| < \tau \\ E_t(x,y) & \text{otherwise} \end{cases} \quad (2)$$

We set the threshold τ to 10cm, which is lower than the height of a human (even while lying on the ground). Figure 4 illustrates example elevation maps of the foreground objects in the real and simulated environments, respectively.

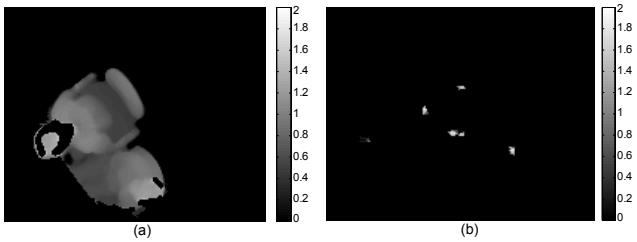

Fig. 4. The elevation maps of (a) the real environment and (b) the simulated environment shown in Figures 2 and 3.

B. Estimation Problems in Full-Resolution Video

After background subtraction, humans can be easily detected in the scene simply by thresholding the foreground map. With the assumption that every person walks into the scene, we threshold the foreground image with a height of T_1 = 0.9m, classifying every connected region (or "blob") above this height as a person. Every person blob in the current frame is compared against the person blobs in the previous frame to find the closest spatial match. We assign the height of the blob to be the elevation value at the weighted centroid [18] of the blob. We denote this height observation in frame t as y_t. We apply a median filter to y_t to mitigate impulsive sensor noise prior to subsequent processing.

Probabilistic models are widely used for activity recognition, since they are well-suited to dealing with noise and uncertainty [19]. Here, we assign a label of "standing" or "sitting" to each person blob using a maximum likelihood estimator that takes into account both the current height observation and a simple Markov model for the probabilities of transitions between states. That is, the label is computed as

$$l_t = \arg\max_l \{p(y_t \mid l) + \lambda \cdot p(l \mid l_{t-1})\} \quad (3)$$

where $l_t \in$ {"stand", "sit"}. The first term reflects the likelihood of the observed height given a proposed label;

for "stand" this is modeled as a Gaussian distribution with mean equal to the observed height of the person on their first appearance. The "sit" distribution is similar, but the mean of the Gaussian is half the entrance height, based on measured data of humans' sitting height ratio [20]. The standard deviations for "sit" and "stand" are 0.1 and 0.2, which were learned from the training datasets described in Section V. The second term $p(l \mid l_{t-1})$ denotes the transition probability, which simply encourages the label to stay the same so that compelling evidence is required in the height observation to change the label. The transition probability matrix we used here is

$$\begin{bmatrix} p(\text{"sit"} \mid \text{"sit"}) & p(\text{"sit"} \mid \text{"stand"}) \\ p(\text{"stand"} \mid \text{"sit"}) & p(\text{"stand"} \mid \text{"stand"}) \end{bmatrix} = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}$$

which was also learned from the training datasets. This makes the pose estimation more robust to noise in the measurement. The parameter λ can be tuned to trade off the prior and measured information; in practice we use a value of 1. The overall pseudo-code for tracking people and estimating their poses is shown in Algorithm 1.

Algorithm 1: Person tracking and pose classification

Input: foreground elevation map $F_t(x, y)$
Output: location, height and pose for each person in the scene

```
for t = 1 to end do
    extract the blobs, record the locations and heights;
    if labeled blobs exist in F_{t-1}(x, y) then
        pass the labels in F_{t-1}(x, y) to the nearest matched
        blobs in F_t(x, y);
    else
        skip this occurrence and wait for new person blob
        entering the scene;
    end
    if new blob without label in F_t(x, y) then
        if at the boundary then
            Assign a new label if higher than T_1;
        else
            erroneous split;
            assign with the nearest label in current frame;
        end
    end
    if blob is labeled then
        classify the pose using (3);
    end
end
```

Due to the low resolution and limited features we can extract from the elevation map, people and pieces of furniture (e.g., moving office chairs) are difficult to robustly segment and distinguish, and often merge into one blob in the elevation map. Thus, the main challenges to this approach are splitting blobs that arise when multiple people are close together, and designing a detector that responds to humans but not to moving furniture (e.g., the back of a chair is often at a comparable height to a seated person's head).

The multi-object merging problem is illustrated in Figure 5a, in which two people P1 and P2 merge into one blob. Here, we describe an approach to split the touching objects, resulting in a subset of local maxima. The basic idea is to process the elevation map to create plateaus with constant intensity around the main peaks. We use a process called morphological reconstruction [18]. The procedure is:

1) Use the original image as a "mask" image;
2) Erode the original image to produce a "marker" image;
3) Repeatedly dilate the marker image until it fits "under" the mask image.

Compared with simple morphological opening, this process tends to "flood" the small maxima, recognize the main local maxima, and preserve the overall shapes of the objects, as illustrated in Figure 6. Without the shape-preserving reconstruction, any maxima, even small ones, will be recognized as local maxima, leading to an overestimate of the number of occupants. Figure 5 shows the procedure of processing the image in Figure 2. We also need to carefully select the structuring elements to erode the images for creating the markers. We use a disk structuring element with a diameter of 7 times the downsampling factor (so that the structuring element is the same effective size regardless of resolution). Figure 7a-b compares simple morphological opening to the result of opening by reconstruction, illustrating the improvement.

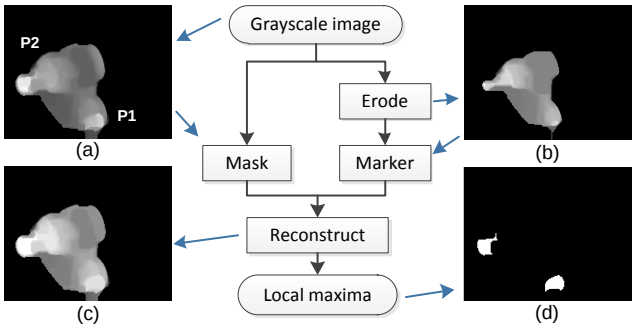

Fig. 5. Extracting local maxima using opening by reconstruction.

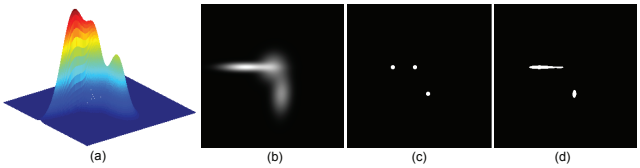

Fig. 6. Finding peaks in blobs created by touching objects. (a) Three Gaussian functions that merge into a single blob when thresholded. (b) The intensity image. The shapes of the local maxima found by (c) simple morphological opening and (d) opening by reconstruction. We prefer the result in (d) since the peak shapes are preserved, and small peaks can be detected and rejected.

However, the proposed method is still susceptible to undesired local maxima. This problem is pronounced when a person leans forward in or gets up from an office chair, generating a new local maximum when the back of the chair is strongly separated from the person's head. This is illustrated in Figure 7c. We address this problem by simply observing that new blobs are only allowed to enter the scene at the edges of the image. That is, a new human cannot spontaneously appear in the middle of the frame. Therefore, we only track the head of the person as they lean forward in the chair and rule out

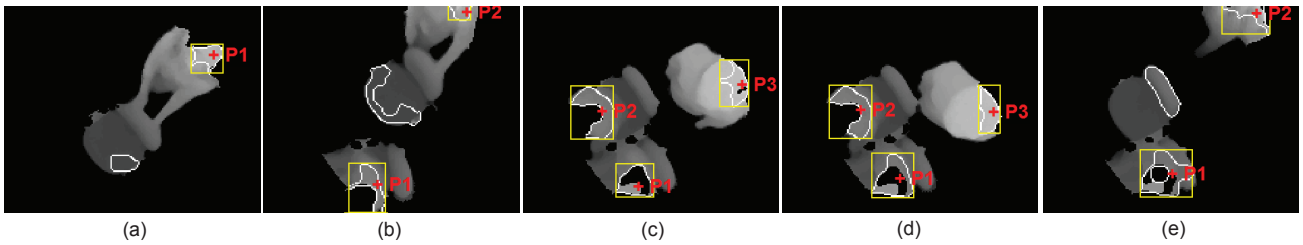

Fig. 8. Multiple person tracking in full-scale frames. Despite the presence of multiple moving office chairs, the correct number of people is maintained.

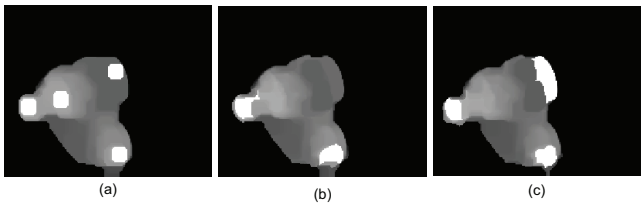

Fig. 7. Extracting local maxima using (a) simple morphological opening and (b) opening by reconstruction from the frame illustrated in Figure 5. In (a), maxima are detected in the wrong places (person P2's shoulder and the back of the chair), while in (b) only the heads are detected. However, in a later frame (c), local maxima extraction using opening by reconstruction finds a maximum corresponding to P2's chair back when P2 leans forward.

spontaneous local maxima. We ensure that in the case of a split, the person label remains with the higher-elevation blob. Conversely, if a blob disappears in the middle of the room (i.e., an instantaneous tracking failure), we store its information to match against future frames.

C. Estimation Problems in Sparsely Sampled Video

The basic algorithm described above works effectively on full-frame images, as illustrated in Figure 8 for several frames. However, our main concern is downsampling the ToF video to simulate widely-spaced single-pixel ToF elements, in order to determine the design requirements for future "smart" ceiling panels. Here, we downsample the full resolution video (using nearest-neighbor sampling) at different spacings to create lower-density video streams, and investigate applying the proposed algorithm to each stream. Figure 9 shows the tracking results in the same frame as Figure 8c for several sensor spacings. Since the downsampled frames are much smaller, the algorithm is much faster and can run at roughly the same rate as the input video.

However, as we push the simulated sparsity to wide spacings, the local maxima extraction algorithm begins to fail due to the limited number of pixels and lack of smoothness in the measurements. In such situations (e.g., a sensor spacing of 25cm), we stop trying to separate multiple people within a blob, and simply keep track of which people are combined in a multi-person blob until they separate (Figure 10). This is expected to suffice for the lighting control application, since maintaining the identity of each person blob is not important.

V. Experimental Results

In this section, we quantify the performance of the detection and pose estimation algorithms as functions of sensor spacing,

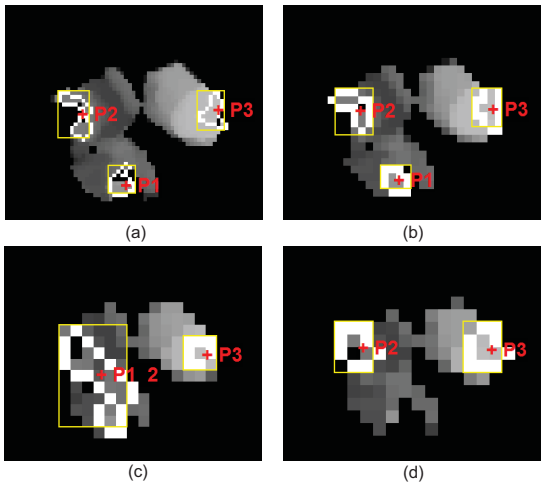

Fig. 9. Privacy-preserving person tracking in the same frame as Figure 8c with different sensor spacing levels of (a)-(d): 0.03m, 0.06m, 0.09m and 0.1m, respectively.

number of people, and type of video.[1] We collected the following datasets for our experiments:

1) real video, 3790 frames, 1–2 people (only used for training)
2) real video, 1050 frames, 1–3 people (only used for testing)
3) real video, 15000 frames, 1–4 people (testing)
4) simulated video, 3000 frames, 6 people (training)
5) simulated video, 2700 frames, 6 people (testing)
6) six simulated videos, 6000 frames each, 5, 10, 15, 20, 25 and 30 people (testing)

A. Experiments on Real Video

We tested the local maxima extraction and tracking algorithm on Datasets 2 and 3. The first video recorded 3 people entering the scene at different times, walking around, sitting on chairs close together, standing up, and leaving. Figure 8 shows one example of tracking on this video.

In every tenth frame of the original elevation video, we manually recorded each person's ID, position, height, and pose. We refer to such manual recordings as "ground truth", and use them to evaluate the performance of our algorithm. Figure 11 shows the absolute errors in location measurements

[1] Videos of the algorithms' results are available at `www.ecse.rpi.edu/~rjradke/tii/`.

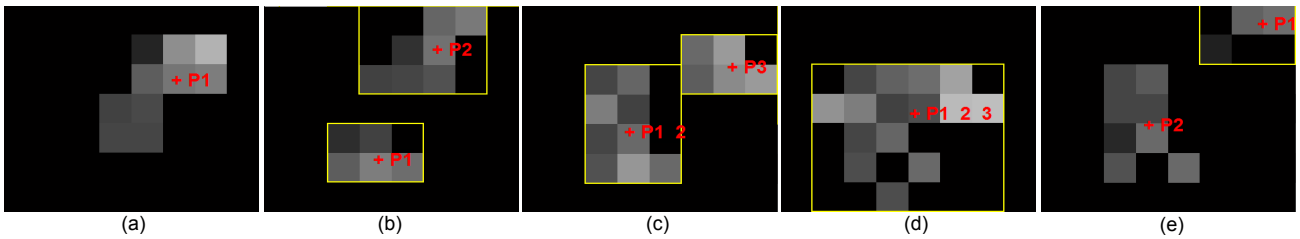

Fig. 10. Privacy-preserving person tracking at a sensor spacing of 25cm. At this level, we don't attempt to separate the people within a multi-person blob.

of the 3 people who enter and leave the scene sequentially in the full-resolution video of Dataset 2. The errors for persons P2 and P3 are higher than for person P1. P2 has long black hair that results in low ToF return intensity, high noise, and missing data (Figure 8c,d). P3 is standing and close to the ToF camera; hence P3's head occupies more pixels. These issues cause the location estimates to be further from the ground truth. In practice, we believe that 10cm location accuracy is sufficient for smart lighting applications. We also noted the general noisiness of the ToF camera, and the spatial non-uniformity of this noise.

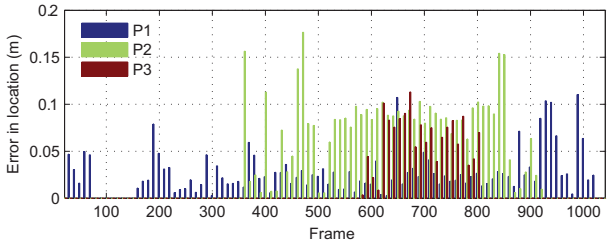

Fig. 11. The errors in location measurements in Dataset 2 with 3 people entering and leaving the scene sequentially.

To analyze the performance at different sensor densities, we spatially downsampled the input video to simulate sensor spacings from 1cm to 10cm in steps of 1cm, as illustrated in Figure 9. At each spacing level, the detections and estimated poses are recorded. Figure 12a shows the error rate of detection and pose estimation, computed as the number of incorrectly detected/missed people (or the number of misclassified poses) divided by the total number of person appearances in all frames. Figure 12b shows the mean error in location and height measurements. We can see that all three people are well-detected and tracked at all spacing levels up to 8cm. When the frames are downsampled to 8cm spacing, the local maxima extraction becomes unstable. The objects are easily missed or merged with each other. Considering Figure 9c and d, in which persons P1 and P2 are both sitting on chairs close to each other, this is not surprising. The algorithm can process 12 frames per second of full-resolution video and 20 frames per second above a sensor spacing of 3cm (i.e., processing frames as quickly as the sensor delivers them).

We also conducted a longer term experiment (Dataset 3) in which the same office environment was continuously monitored for 15000 frames. During this time, each person entrance/exit and pose change (e.g., sitting to standing) was

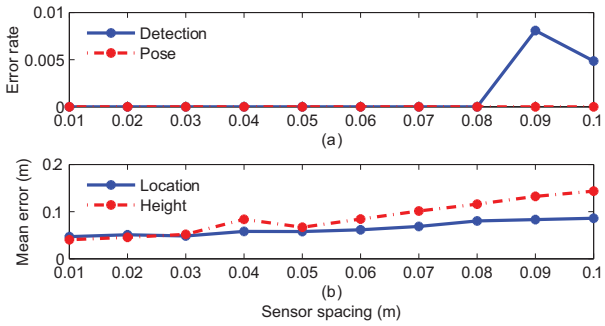

Fig. 12. Performance at different sensor spacing levels from 1cm to 10cm in steps of 1cm. (a) Error rates in detection and pose classification. (b) Mean errors of location and height measurements.

TABLE I

PERFORMANCE COMPARED WITH GROUND TRUTH FOR EXPERIMENTS ON REAL VIDEO

	Ground Truth	Detected	False Alarm
Entrance	31	31	2
Exit	31	31	2
Stand up	54	53	2
Sit down	53	50	3

manually recorded, using the full-frame video for ground truth. The algorithms were run in real time on the 4cm spacing level video and compared with the ground truth. In the ground truth, there were 31 entrance and exit events each, and 54 total stand up and 53 sit down events. The algorithm running on the downsampled video successfully detected all of the entrance and exit events, and almost all of the pose change events, with only a few false alarms. The results are summarized in Table I.

B. Experiments on Environmental Simulation

As discussed in Section III, we simulated a large 18×14 m^2 lab environment using the Unity game engine. To test our privacy-preserving tracking algorithm, we recorded a video containing 2700 frames (Dataset 5), involving six people entering the lab, walking around, sitting down, standing up, standing close to each other, grouped together around the conference table and in front of the experimental bench, and leaving the room. The elevation maps can be directly obtained by the game engine, and processed into video that mimics the output of the actual time-of-flight sensor. To reflect the observed noise in the SR4000 sensor, we added Gaussian

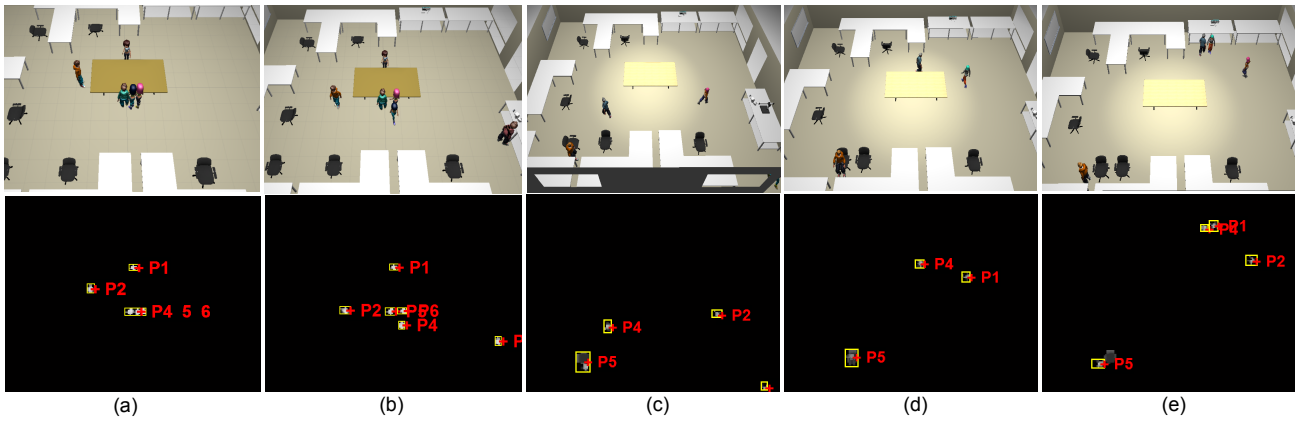

Fig. 13. Snapshots of the scene and tracking result in full-scale frames for the simulated experiment.

random noise with standard deviation $\sigma = 4$ mm to every pixel in the simulated data. Figure 13 shows several snapshots of the simulation, which include several challenging situations. In Figure 13a-b two or three people merge into a blob and later split apart; in Figure 13c-e person P5 moves with a chair, sits on the chair and then leaves the chair. The ground truth of each person's identity, location, and pose are directly recorded from Unity.

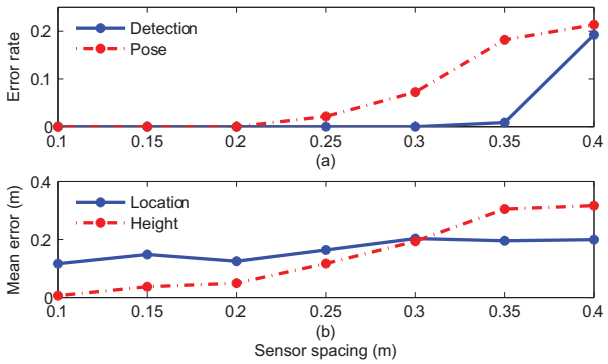

Fig. 14. Performance for the simulated experiment at different sensor spacing levels from 10cm to 40cm in steps of 5cm. (a) Error rates in person detection and pose classification. (b) Mean errors of location and height measurements.

As before, we gradually increased the sample spacing, this time from 10cm to 40cm in increments of 5cm. Figure 14 reports the error rate in person detection, pose classification, and location estimation. The error rate and estimation errors are acceptably low up to a spacing level of about 35cm, at which point people are frequently lost between sensors and can easily become mislabeled. That is, above this spacing level, a person can fit "in between" two ToF rays, leading to an inaccurate measurement of their height. At all sensor spacings, the algorithms run at about 30 fps on the synthesized video.

Table II reports the detection and false alarm rates for entrance, exit, sit down and stand up events at the 30cm spacing level, indicating excellent performance.

TABLE II
PERFORMANCE COMPARED WITH GROUND TRUTH FOR EXPERIMENTS ON ENVIRONMENTAL SIMULATION

	Ground Truth	Detected	False Alarm
Entrance	10	10	0
Exit	10	10	0
Stand up	10	10	1
Sit down	10	10	1

TABLE III
PERFORMANCE COMPARED WITH GROUND TRUTH FOR EXPERIMENTS ON DIFFERENT NUMBERS OF PEOPLE IN REAL DATA

Number of people	1	2	3	4
Detection error rate (%)	0	0	0.16	0.82
Location mean error (m)	0.10	0.11	0.12	0.12

C. Performance With Respect to Person Density

Finally, we analyzed the performance of the proposed algorithms with respect to the number of people in the frame, for both the real and simulated datasets.

In the real case, we studied the long video in Dataset 3, using a large sensor spacing of 8 cm. Table III reports the results, breaking down the detection error rate and location error with respect to the number of people in the frame at each instant. As expected, the detection error rate is perfect (i.e., 0) when only one or two people are in the frame, and only increases slightly (still less than 1%) as the frame becomes more crowded. The occasional errors are due to lost local maxima inside multi-person blobs. The location error for detected objects stays roughly constant at about 10cm (which makes sense given the human head radius). Since only four people can fit into the field of view of the real sensor, we now proceed to discuss our simulated environment, which can hold many more people.

In the simulated case, we can populate the environment with many more people, from 5 to 30 (Dataset 6). We used a wide sensor spacing of 20cm for this experiment. Table IV reports the results. Again, the detection error rate is 0 up to a certain level of crowding (15 people), at which point occasional errors are made. Even for 30 people, the detection error rate remains

TABLE IV
PERFORMANCE COMPARED WITH GROUND TRUTH FOR EXPERIMENTS ON DIFFERENT NUMBERS OF PEOPLE IN SIMULATION

Number of people	5	10	15	20	25	30
Detection error rate (%)	0	0	0	0.02	0.53	0.72
Location mean error (m)	0.11	0.10	0.11	0.11	0.11	0.12

below 1%. As in the real data, we found the location error for detected objects to be roughly constant at about 11cm.

VI. CONCLUSIONS AND DISCUSSION

We described a method for accurate, real-time person tracking and coarse pose recognition using a ToF camera that preserves the privacy of a room's inhabitants. These results will inform the design of future smart ceiling panels, as well as adaptive lighting control algorithms in smart rooms. While single-pixel ToF sensors are not yet commercially available, we expect they can be designed at low cost in the near future using methods based on nanoplasmonic devices [21].

We observed that while the local maxima extraction works well on the full-scale images, objects can be lost in the downsampled images due to occlusions or morphological processing. This reflects a natural tradeoff between the competing goals of high accuracy and privacy preservation. The user could tune the downsampling rate to correspond to a comfortable tradeoff.

The current Markov model acts as a temporal smoothing, to avoid "jitter" during situations when a person is bent over at approximately the sit/stand threshold, but this could be improved by considering more than just the previous frame. Since lighting control algorithms need not operate at video frame rates, we would probably be willing to tolerate increased processing time to obtain increased accuracy. Depending on the sensor spacing, we also plan to investigate further heuristics for reasoning about locations and heights in multi-person blobs.

Our next investigations in this area involve extensions of the Unity game engine to create larger and more realistic environment simulations (e.g., a student union or a multi-floor office building). This requires not only larger-scale building models but also more accurate behavior simulations for the simulated human participants. An alternate approach is to integrate simulated time-of-flight measurements into research tools such as the virtual environment simulator proposed by Starzyk and Qureshi [22].

Finally, we plan to integrate the occupancy and pose detection framework described here with actual lighting control systems, to investigate strategies for saving energy and improving human comfort and task performance.

REFERENCES

[1] P. Waide and S. Tanishima, *Light's labour's lost: policies for energy-efficient lighting*. OECD/IEA, 2006.

[2] G. Augenbroe and C. Park, "Quantification methods of technical building performance," *Building Research & Information*, vol. 33, no. 2, pp. 159–172, 2005.

[3] R. A. Jarvis, "A laser time-of-flight range scanner for robotic vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, pp. 505–512, 1983.

[4] A. Kolb, E. Barth, R. Koch, and R. Larsen, "Time-of-flight cameras in computer graphics," in *Computer Graphics Forum*, vol. 29, no. 1, 2010, pp. 141–159.

[5] Y. Cui, S. Schuon, D. Chan, S. Thrun, and C. Theobalt, "3D shape scanning with a time-of-flight camera," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[6] L. Bianchi, R. Gatti, L. Lombardi, and P. Lombardi, "Tracking without background model for time-of-flight cameras," *Advances in Image and Video Technology*, pp. 726–737, 2009.

[7] S. Guðmundsson, M. Pardas, J. Casas, J. Sveinsson, H. Aanæs, and R. Larsen, "Improved 3D reconstruction in smart-room environments using ToF imaging," *Computer Vision and Image Understanding*, vol. 114, no. 12, pp. 1376–1384, 2010.

[8] S. Gokturk and C. Tomasi, "3D head tracking based on recognition and interpolation using a time-of-flight depth sensor," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[9] S. Malassiotis and M. Strintzis, "Robust real-time 3D head pose estimation from range data," *Pattern Recognition*, vol. 38, no. 8, pp. 1153–1165, 2005.

[10] D. Hansen, M. Hansen, M. Kirschmeyer, R. Larsen, and D. Silvestre, "Cluster tracking with time-of-flight cameras," in *CVPR Workshop on Time of Flight Camera Based Computer Vision*, 2008.

[11] Q. Cai, D. Gallup, C. Zhang, and Z. Zhang, "3d deformable face tracking with a commodity depth camera," in *European Conference on Computer Vision*, 2010.

[12] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, "Real time motion capture using a single time-of-flight camera," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[13] *SwissRanger SR4000 user manual*, MESA Imaging AG, 2010.

[14] T. Oggier, M. Lehmann, R. Kaufmann, M. Schweizer, M. Richter, P. Metzler, G. Lang, F. Lustenberger, and N. Blanc, "An all-solid-state optical range camera for 3D real-time imaging with sub-centimeter depth resolution (SwissRanger)," in *Proc. SPIE*, vol. 5249, no. 65, 2004, pp. 534–545.

[15] S. Guðmundsson, H. Aanæs, and R. Larsen, "Environmental effects on measurement uncertainties of time-of-flight cameras," in *International Symposium on Signals, Circuits and Systems*, 2007.

[16] Unity game engine. [Online]. Available: http://www.unity3d.com

[17] P. Soille, *Morphological image analysis: principles and applications*. Springer-Verlag New York, Inc., 2003.

[18] R. Gonzalez, R. Woods, and S. Eddins, *Digital image processing using MATLAB*, 2nd ed. Gatesmark Publishing, Gatesmark, LLC, 2009.

[19] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *Computer Vision and Image Understanding*, vol. 115, no. 2, pp. 224–241, 2011.

[20] A. Fredriks, S. van Buuren, W. Van Heel, R. Dijkman-Neerincx, S. Verloove-Vanhorick, and J. Wit, "Nationwide age references for sitting height, leg length, and sitting height/height ratio, and their diagnostic value for disproportionate growth disorders," *Archives of disease in childhood*, vol. 90, no. 8, p. 807, 2005.

[21] G. Konstantatos and E. H. Sargent, "Nanostructured materials for photon detection," *Nature Nanotechnology*, vol. 5, pp. 391–400, 2010.

[22] W. Starzyk and F. Qureshi, "A distributed virtual vision simulator," in *Conference on Computer and Robot Vision*, 2012.

Li Jia Li Jia is currently working towards a Ph.D. degree in Computer and Systems Engineering in the Department of Electrical, Computer, and Systems Engineering at Rensselaer Polytechnic Institute. She received a B.S. degree in Electrical Engineering from Nanjing University of Aeronautics and Astronautics in Nanjing, China in 2005 and a M.S. degree in Electronic Information Engineering from Beihang University in Beijing, China in 2008. She worked as a software engineer in the China Academy of Space Technology in Beijing, China in 2008-2009. Her research interests include depth sensing, multi-object tracking and environmental simulation for smart lighting control systems.

Richard J. Radke Richard J. Radke joined the Electrical, Computer, and Systems Engineering department at Rensselaer Polytechnic Institute in 2001, where he is now an Associate Professor. He has B.A. and M.A. degrees in computational and applied mathematics from Rice University, and M.A. and Ph.D. degrees in electrical engineering from Princeton University. His current research interests include computer vision problems related to modeling 3D environments with visual and range imagery, designing and analyzing large camera networks, and machine learning problems for radiotherapy applications. Dr. Radke is affiliated with the NSF Engineering Research Center for Smart Lighting and the DHS Center of Excellence on Explosives Detection, Mitigation and Response (ALERT). Dr. Radke is a Senior Member of the IEEE and an Associate Editor of *IEEE Transactions on Image Processing*. His textbook *Computer Vision for Visual Effects* was published by Cambridge University Press in 2012.