Inti: Indoor Tracking with Solar Cells

Oxana Oosterlee
Technical University Delft
o.oosterlee@tudelft.nl

Talia Xu
Technical University Delft
m.xu-2@tudelft.nl

Marco A. Zúñiga Zamalloa
Technical University Delft
m.a.zunigazamalloa@tudelft.nl

Abstract

Solar cells are mainly used as power sources, but can be used for sensing as well. We propose a novel indoor system that exploits solar cells to track people by monitoring the changes in light intensity caused by their shadows and reflections as they walk by. Our framework has three main components. First, we develop a simulator based on a ray-tracing model to determine how the solar cells should be positioned in the tracking environment to maximize the signal to noise ratio. Next, we apply changepoint detection methods to convert the (noisy) solar cell signal into a binary detection signal. Our detection method uses a Bayesian approach, which allows our system to work well in various environments, with natural and artifical light. Finally, the binary output from multiple solar cells is fused to track multiple targets. The tracking engine is based on a particle filter implementation based on the probability hypothesis density filter. This approach allows us to perform tracking without knowing the actual number of targets in the environment. To evaluate our framework, we build small tags that consist of a solar cell, a micro-controller and a wireless module, and deploy them in a real apartment. Ours results show that our system allows solar cells to track people under different lighting conditions, during day and night.

1 Introduction

Indoor tracking is an active research area with a wide variety of applications, ranging from elderly care to occupancy monitoring. Contrary to indoor positioning, where fine-grained information is required, indoor tracking only requires a coarse-grained sequence of locations to determine trajectories.

A common property of most indoor tracking systems is that they are *passive*, which means that targets do not need to carry any device. Multiple sensing modalities have been explored to achieve such passive approach. Cameras and acoustic sensors provide reliable tracking but they raise privacy concerns [6]. To overcome these privacy-issues, researchers have proposed the use of radio tomography [15], seismic sensors [2], and light sensors [8]. These sensing modalities can detect the presence of a target but cannot identify it. Following this type of privacy-preserving tracking, our study proposes a novel indoor tracking system that exploits ambient light. Similar to some prior studies [7, 16, 5, 8, 24], our system leverages the shadows and reflections caused by people as they walk by, but contrary to those studies we do not modify the lighting infrastructure [7, 16, 23], utilize complex training phases [5, 8], or require placing the sensors on rather inconvenient places such as the floor [24].

Aim. We envision a system that is not only passive and privacy-preserving but also easy to deploy and maintenance-free. The aim is to deploy small tags consisting of solar cells, which can act not only as energy harvesters but also as photosensors. The tags will report the presence (or absence) of people to a central location, which will be in turn responsible to merge all the data to obtain the final trajectories. For this work, we are focusing on demonstrating the capabilities of solar cells as sensors to track movements in an indoor setting. We aim to build an end-to-end tracking system using solar cells to detect and track passing targets.

Challenge. At first glance, the problem may look deceptively simple. Solar cells (or light sensors in general) can provide information that any tracking algorithm could process to obtain trajectories. The challenge arises due to the complex interaction between the type of light source (artificial or natural), the reflective properties of the surrounding surfaces (environment and people), and the distance between the people passing by and the sensors. All these variables create a highly variable and noisy environment.

Contributions. Considering the above challenge, our system, dubbed Inti, provides the following contributions.

section 2: Placement. To facilitate target detection, the tags need to maximize the changes that shadows and reflections cause on the solar cell signal. But an indoor setup offers multiple locations where the tag can be placed, what is the optimal location, height and angle to maximize those signal changes? We propose a ray-tracing approach to identify the optimal placement of our solar cell tags.

section 3: Detection. The changes on the received signal are not simple valleys or peaks created by the shadows and reflections, there are multiple extrema with different intensities. Simple peak detection methods would not work. We adapt and demonstrate that a Bayesian approach [1] is well-suited to detect events without the need to fine-tune parameters for every specific scenario.

section 4: Tracking. Even with the best placement and detection methods, the sensors will provide false positives and false negatives due to the complex propagation of light waves. Among various tracking algorithms, we adapt an implementation of Particle Filters based on Probability Hypothesis Density [13] and demonstrate its feasibility to overcome misreadings in multi-target scenarios.

section 5: Evaluation. To demonstrate the validity of our framework, we build a simple platform consisting of solar cells, a microprocessor and BLE transmitter, and deploy six of these tags in a small apartment. Our evaluation considers multiple types of trajectories with different lighting conditions, during the day with sunlight and during the night with artificial lighting.

2 Simulation

To detect and track a target with solar cells, the target first has to cause sufficient variations on the amount of light reaching the tag. In this section, we describe a simulation tool that we developed to identify the best placement options for the tags.

Light-based sensing has been studied extensively. In several interactive applications, participants utilize photo sensors at very close distances, of a few centimeters or less [11, 10]. In this benign setup, most of the light reaching the sensor can be blocked, creating clean signals. In our study, however, the targets move at longer distances from the solar cells (around one meter or more), resulting in smaller and noisier signals. This makes the detection and tracking a more difficult task. This difficulty is further compounded by the nature of lighting in an indoor environment, where multiple light sources are typically present to ensure uniform lighting, and walls and furniture have different reflecting and absorbing properties. Such a setup generates multiple paths for the light rays and results in additional noise on the solar cell output.

In an uncontrolled and noisy environment, with many light sources and a limited number of solar cells, it can be difficult to determine the desirable locations to place the tags. The solar cells could be placed on the ceiling, floor or along the walls. Among these three options, installations along walls is the most suitable because placements on the floor are impractical and placements on the ceiling reduce the received signal strength coming from reflections bouncing off the floor. Installations along the walls, however, pose multiple options. For example, the solar cells can be placed on different heights and have different angles. In this section, we develop a simulation tool to identify the height and angle of solar cells that maximize the (light) changes caused by passing people. To simulate the voltage levels observed by solar cells over time, our tool considers different solar cell placements and different walking paths of a moving target.

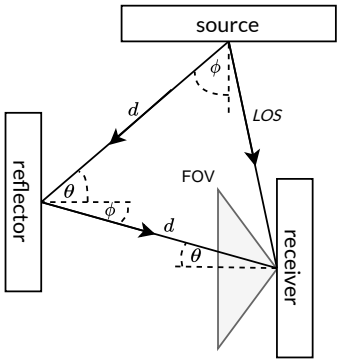


Figure 1: Light rays from a source to a receiver. Both types of path, LOS and reflections, are depicted.

A higher voltage variation corresponds to a better configuration, since the signal will be easier to process.

2.1 Ray-tracing

The simulator is based on the concept of ray-tracing. Ray-tracing is a method that models the behaviours of light rays and their interactions with other objects in an environment [17]. Three types of objects are defined in our ray-tracing method, as shown in Figure 1: A ***source*** represents the Lambertian light sources that emit light, such as a light bulb. A ***receiver*** represents the solar cells, which collects the incoming light rays. A ***surface*** represents the diffuse and reflective surfaces in an environment, such as walls, floors and doors.

In this ray-tracing method, light emitted from the source is represented by a set of individual light rays, each carrying a portion of the power of the light. Each ray propagates in the environment in different directions, and can be reflected or absorbed when it encounters different surfaces. As a target (person) appears in different locations, it can *block off* the paths between the receivers and the light source, but can also *reflect* additional light rays into the receiver. The power of light measured by the solar cells is approximated by summing up the number of light rays impinging over them.

2.2 Applying Ray-tracing for Tracking

In the previous section, we describe the ray-tracing algorithm in a static environment. In a dynamic setup, as a target passes by, the ray-tracing process is repeated at each time step to estimate the changes in voltage output observed on the solar cell. This simulation process is depicted in Fig. 2 and it follows four main steps:

Step 1: We generate an *approximate* 3D model of the environment. This model captures the location of the light and the dimension of the space (floor, ceiling and walls).

Step 2: We choose various solar cell positions over the wall, with different heights and angles.

Step 3: A human target is simulated, and moves through the environment along a predefined path. The paths go from one end of a passageway to the other, at different distances from the walls. At each time instance dt, the target moves forward by a distance d and the ray tracing algorithm is executed. For each time instance, the expected received power for each solar cell is calculated and recorded.

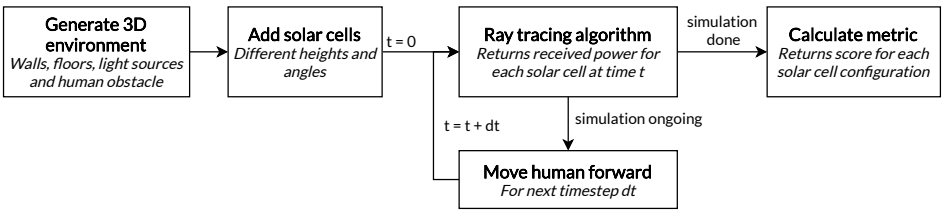


Figure 2: Step-by-step overview of the simulation process for determining the best solar cell configuration.

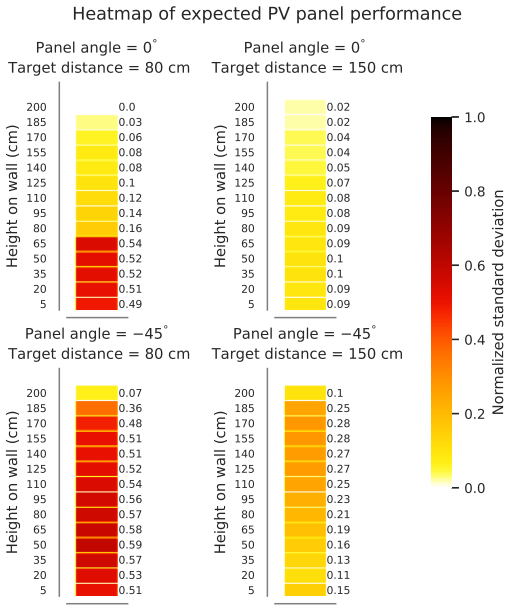
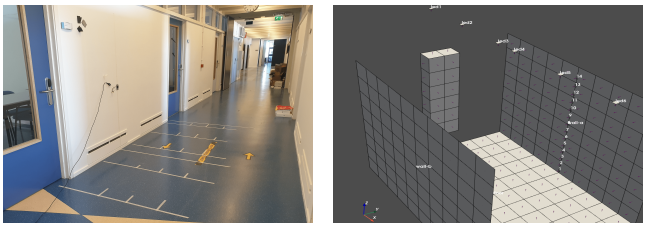


Figure 3: Heatmap

Step 4: After the target has finished moving along the path, the simulation run is finished.From this data, we evaluate the quality of each tag placement by calculating the standard deviation of the voltage output over the simulation run.

Fig. 3 shows an example of the heatmaps that describe how the standard deviation of the received light intensity changes for different solar cell positions, angles, and target distances. Two sample angles are shown, 0°, and −45° (facing towards the floor). The difference between the left and right plots is the distance at which the target passed the solar cells, 80 cm and 150 cm respectively.

It can be seen that in general, a solar cell at an angle of −45° is expected to perform better than one at 0°, as it results in a higher standard deviation. In addition, we can see that cell configurations performing best for targets at a close distance do not necessarily perform best for targets passing at a longer distance. For example, when positioning the solar cell at −45°, placing the tag at a height of around 50 cm gives the highest detection rate for a target passing at 80 cm distance. However, when the target passes at a distance of 150 cm, this configuration does not give the best result. Instead, for this longer range the solar cell should be positioned around 155 cm. These results illustrates how the simulator can identify the best placements for a given range.



(a) Picture of the hallway during the experiments (with only one solar cell).

(b) Screenshot of the hallway in simulation.

Figure 4: Picture of the real hallway and the hallway in simulation (no rays traced).

Table 1: Normalized standard deviation calculated from the simulation output for selected solar cell configurations. The target distance y is in cm.

				Variance for target distance y			
	Rating	Height	∠	y=50	y=80	y=110	y=150
Solar cell config.	Good	50 cm	-45°	0.95	0.61	0.31	0.15
		35 cm	-25°	1.00	0.63	0.31	0.18
	Med	110 cm	-10°	0.80	0.15	0.13	0.08
		35 cm	0°	0.88	0.58	0.15	0.08
	Med-L	140 cm	-5°	0.79	0.09	0.08	0.04
		125 cm	0°	0.81	0.09	0.08	0.05
	Bad	200 cm	-45°	0.00	0.05	0.11	0.09
		185 cm	-10°	0.01	0.02	0.03	0.02

2.3 Validation

In this section, we perform empirical measurements to validate the simulation trends. The setup used for both, the experiments and simulation, is a hallway in an office setting, as shown in Fig. 4. Since we only *approximate* the environment in the simulation, our goal is not to obtain the exact measurements at the tags, but to identify if one tag configuration (location and angle) is better than another. In other words, considering two tags, i and j, if tag i provides a higher voltage than tag j in the real setup, we want the simulator to also show that tag i is better, even if the simulated and empirical voltage values do not match.

Simulation setup. The dimension of the hallway and the locations of the lights are the same as in the real setup. A set of solar cell configurations with different heights and angles are evaluated for different paths. For each path, a target – modeled as a big block– moves along the hallway. Based on the results of the simulator, we rate the quality of each solar cell configuration to be one of ”Good”, ”Med (medium)”, ”Med-L (medium-low)”, and ”Bad”.

Selected configurations. The simulation results evaluate a

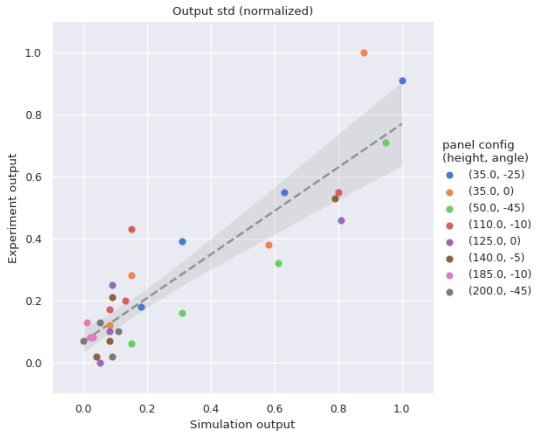


Figure 5: Comparison of the simulation and experimental results.

large number of configurations, but measuring all of them in a real setup would be time consuming. To validate the correctness of the simulation results, we select eight configurations –two in each quality rating category (Good, Med, Med-L, Bad)– as shown in Table 1.

Experimental setup. For each tag configuration (location and angle), we perform experiments with the same walking paths to compare the simulated and empirical results.

Fig. 5 shows the normalized results of both the simulated and the experimental values for the eight configurations. For each configuration, we consider four paths with different distances from the wall: 50 cm, 80 cm, 110 cm and 150 cm. Hence, each configuration –represented by a different color– has four points in the figure. The dashed line indicates a perfect agreement between simulation and experimental results. The closer a data point is to the dashed line, the closer the results are between simulation and experiments.

It can be observed in Fig. 5 that there is a good agreement between the ranking given by the simulation and the experiments, as the data points concentrate near the dashed line. More importantly, what the simulator identifies as *bad* placements are also poor placements with the empirical results. Note that the two bottom configurations (pink and gray dots) are concentrated at the bottom left. For these placements, the variation of the voltage output is small both in the experiment and simulation, making it difficult to correctly detect passing targets. On the other hand, the two configurations rated as *good* in Table 2 (green and blue dots) extend to the top right part of the figure. The placements rated as medium (orange and red) and medium-low (purple and brown) only perform reasonably well when the target passes nearby the wall, but with more erratic correlations for other paths.

Even though a one-to-one correspondence between the simulated and experimental results is difficult to obtain, we can observe from Fig. 5 that we can still achieve useful insights. From the validation results, it can be seen that the simulator can effectively inform the placements of solar cells for target detection. In the experimental environment, the hallway has both *artificial light*, which is captured in the simulator, as well as *ambient light* from the window, which is not captured. Despite the fact that the simulation setup does not

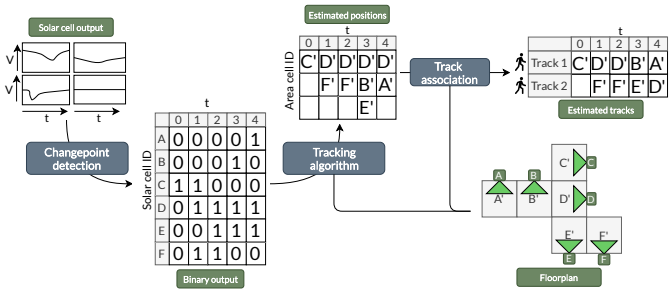


Figure 6: Overview of the steps required for multi-target tracking with example data.

capture everything in real life, we were able to get a good agreement in terms of the quality of solar cell configurations between the simulation and experiments. In the evaluation we perform in the real apartment (Section 5), we deploy the tags using the guidelines obtained in this section.

3 Detection

Once we identify a good placement for the tags, we need to use these signals over time to detect passing targets. An overview of the steps involved in target tracking is shown in Fig. 6. The analog output of the solar cells at each timestep go through a changepoint detection algorithm, which produces a binary output indicating whether a target has been detected. This binary output is then processed in a tracking algorithm (explained in the next section) to determine the paths followed by people.

In this section, we describe how the changepoint detection algorithm is able to detect the presence of targets from solar cell outputs. The presence of ambient light makes it a more challenging task than simply identifying the valleys in a signal with a thresholding algorithm. We describe the adaptation of a Bayesian method to mitigate the effects of the changing lighting conditions throughout the day. Later, we show that this Bayesian approach outperforms a thresholding algorithm under varied experimental conditions.

As a person moves across the room from one side to another, the variations on the solar cell outputs can have different characteristics. An example of six solar cells placed in different locations in a room as a person walks by can be seen in Fig. 7. In the cases of solar cells C and F, pronounced dips in the signal can be observed as a person walks by. However, this is not true for every solar cell. For example, in the case of solar cell D, a peak is observed instead of a dip due to reflections of light. In the case of solar cells A, B and E, many small fluctuations are observed instead. The detection algorithm is able to identify the changes both when they are pronounced (C, D and F), and when there are small fluctuations (A, B and E), as shown with red dots. On top of this, the level of ambient light changes overtime, posing additional challenges in associating changes in solar cell outputs to passing targets.

To address the challenges in detecting a passing target from the noisy signals, we need to design a detection algorithm that can meet the following requirements: First, the detection algorithm should be able to adapt to the changing ambient light conditions. The detection algorithm should

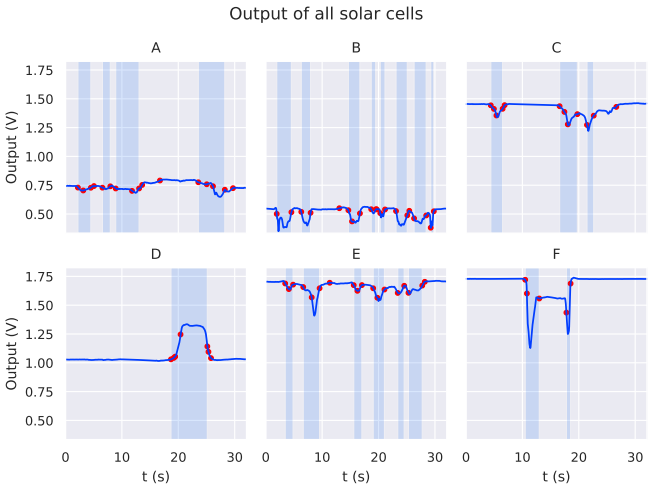


Figure 7: Variations of solar cell outputs as a target walks by.

be able to achieve this without prior training, otherwise, the fine-tuning process has to be repeated for every scenario and has to change between environments. For example, the intensity of the ambient light can vary significantly throughout the day, and the furniture and objects present in a room can change from environment to environment, and also overtime. The detection algorithm should be able to adapt to these changes without requiring training. Second, the algorithm should be able to detect passing targets generating different types of changes on the solar cell outputs, peaks and valleys. Third, to connect the target detection phase with the target tracking phase (explained in the next section), the detection algorithm has to be able to determine whether a target is present in real time. Therefore, it should be able to identify the changes in the solar cell outputs using only the received data from the past.

For these reasons, we consider a Bayesian approach called the Bayesian online changepoint detection (BOCD). The BOCD algorithm constantly estimates and updates the mean and variance of the probability distribution of a signal [1]. As new measurements arrive from the solar cell outputs, the BOCD algorithm estimates whether the new data points belong to the same probability distributions of previously received data points. This approach compensates for the slow changes in the environment, for example, the changes in daylight will lead to a smooth adaptation of the underlying distribution. In the meantime, when a target passes a solar cell, causing the output to change significantly in a very short period of time, the BOCD algorithm will recognize that this change comes from a new data distribution.

In the BOCD algorithm, when the received data points start to exhibit different characteristics to previously received data points, a changepoint event occurs. For example, in Fig. 8, events are detected a times 3 and 5. A general description of the steps in a BOCD algorithm is shown in Algorithm 1, while the detailed algorithm can be found in [1]. With the default parameters, as presented in [1], the BOCD algorithm has two important features. First, a changepoint happens at time n when the probability of a new data point

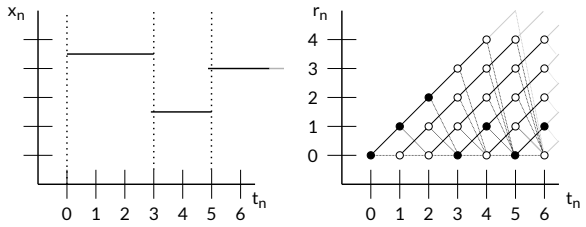


Figure 8: Evolution of the value for run length r_n for a signal x_n with changepoints at t=3 and t=5. Left: The data point received at each timestep. Right: The evolution of the run length r_n. For each timestep, the value of r_n increases by one (solid line), or it is reset to $r_n = 0$ when a changepoint occurs (dashed line). The value of r_n that is most likely for the timestep is colored black.

$P(r_n = 0|\mathbf{x}_{0:n})$ exceeds a threshold $P_{r,min}$ (time stamps 3 and 5 in Fig. 8 right). Second, changepoints can occur at consecutive time stamps. In our empirical experiments, we found that these default features do not provide good results and give many false positives (FPs) and false negatives (FNs). Next, we explain in more detail the reasons for these undesirable phenomena and the solutions we provide.

- FPs: As targets pass by a solar cell, the changes they cause are not consistent and may include intermittent transitions, as shown in solar cell B in Fig. 7, which can cause FPs in the detection results. This occurs because the BOCD algorithm uses a variable called the run length r to keep track of the time between changepoint events. Fig. 8 shows an example of how the value of r is updated. At $t_n = 0, 3, 5$, changepoints are detected, which cause r_n to be set to 0. At other time steps, r_n is incremented by one. BOCD has no limits on the minimum value of r: changepoints can occur consecutively, which is a problem for our noisy signals because a single event is detected as multiple (consecutive) events. To reduce the number of FPs, we define a minimum time (run length) between two consecutive changepoints. This minimum time is determined by the sampling rate of the system and considers that people require a few seconds to cross over the field-of-view of the solar cell.
- FNs: At each time step, if the BOCD algorithm determines that the probability of the new data point x_n is higher than the threshold $P_{r,min}$, the point is considered a 'changepoint event', which indicates that a change of the underlying distribution has occurred. The problem is that, depending on the illumination environment and the followed paths, the changes caused by passing targets are not always instantaneous. Therefore, it can be difficult for the BOCD algorithm to determine the exact timestep when a changepoint event occurs, leading to False Negatives. To reduce the number of FNs, instead of applying a threshold $P_{r,min}$ on a *single* timestep, we apply the threshold over a predetermined *range* of adjacent timesteps. This approach allows us to identify moderate changes in probability, without missing sud-

Algorithm 1: The BOCD algorithm of a signal with Gaussian distribution

At each timestep n,

1. Observe a data point x_n
2. Calculate the possibility that x_n belongs to the currently estimated method
   $\pi_n{}^{(r)} = P(x_n|\mu_n{}^{(r)}, \sigma_n^{2(r)})$,
   where (r) indicates the current run, μ and σ^2 represents the mean and variation of the estimated Gaussian distribution, respectively.
3. π_n is used to calculate the growth probability and changepoint probability
   $P(r_n = r_{n-1}+1, \mathbf{x}_{1:n}) = P(r_{n-1}, \mathbf{x}_{1:n-1}) \cdot \pi_n^r \cdot (1 - H(r_{n-1}))$,
   $P(r_n = 0, \mathbf{x}_{1:n}) = \sum_{r_{n-1}} P(r_{n-1}, \mathbf{x}_{1:n-1}) \cdot \pi_n^r \cdot H(r_{n-1})$
4. The probability distribution of the runlengths are normalized, if $P(r_n = 0, \mathbf{x}_{1:n})$ is larger than a predefined threshold P_{r_min}, the runlength is reset to 0. Otherwise the runlength is incremented by one
5. The mean and μ and variance σ^2 of the distribution is updated according to the methods described in [14]

den transitions. Formally, in our system, a changepoint is detected when $\sum_{r=\hat{r}-n_d}^{r=\hat{r}+n_d} R_{n,r} \geq P_{r,min}$, where n_d is a predefined range.

3.1 Evaluation

To evaluate the performance of the BOCD algorithm, we apply it to the experimental data obtained for the hallway in Section 2.3 (32 test points in total). In the experiments, the target position at each timestep was recorded, making it possible to provide a ground truth for analysis.

To evaluate the accuracy of the detection algorithms, we need to determine at what timesteps the target is considered detectable by a solar cell. For this, we first calculate the distance of the target to the tag for each timestep:

$$d_{x_t} = |x_{receiver} - x_{human,t}|, \tag{1}$$

$$d_{y_t} = |y_{receiver} - y_{human,t}|, \tag{2}$$

$$d_{s_t} = \sqrt{d_{x_t}^2 + d_{y_t}^2} \tag{3}$$

where d_{x_t} indicates the distance between a passing target to the wall where the solar cell is placed. Considering the field of view of the solar cell, we consider a target within the detectable range of a receiver when d_{x_t} is less than 40 cm and d_{s_t} is less than 100 cm. We compare the performance of the BOCD algorithm with a simpler, gradient-based threshold algorithm, dubbed GCPD, which determines whether a target is present by calculating the gradient of the data within a time window and comparing it against a pre-defined threshold. Instead of using the accuracy, we use the F1-score in the comparison, which is a commonly used metric in statistical analysis of binary classification [20]. The F1 score is more advantageous in considering the effects of FPs and FNs, as well as when the number of data points in each category is very different. The F1 score is given by:

$$\text{F1} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{4}$$

Figure 9 shows a violin plot for the F1-scores for both BOCD and GCPD. The average F1-score is similar for both algorithms. For BOCD it is 0.56 (average precision: 0.47, average recall: 0.85). For GCPD it is 0.51 (average precision: 0.47, average recall: 0.68). However, BOCD is much more consistent in its performance than GCPD. The scores of BOCD are spread evenly around the average value. On the other hand, GCPD has a larger number of runs performing very well (F1-score larger than 0.75), but on the other hand also has a significant number of runs scoring very poorly (near 0). BOCD scores significantly better in recall, meaning that it is better at detecting the existence of changepoints.

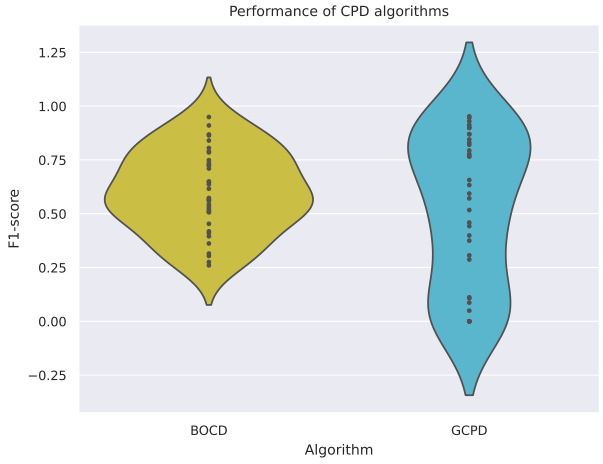


Figure 9: Violin plot of F1-scores of the BOCD and GCPD algorithms used for changepoint detection. Each point shows the score of one of the 32 experiment runs.

In conclusion, for well-controlled scenarios, where targets pass by the solar cells at approximately the same distance, and no change in the environmental condition exists, GCPD can be fine-tuned to outperform BOCD. However, BOCD is more robust and gives a better overall performance when changes occur in the experimental setup and when experiments are carried out in different environments.

4 Tracking

In the previous section, we applied changepoint detection methods to produce binary outputs from the received solar cell signals. At each time step, the detection outputs indicate whether a human target is present. Our goal in this section is to track the walking paths of human targets in an indoor area

by combining the binary outputs of multiple solar cells.

However, we face two main challenges. First, factors such as changes of light conditions in the environment and different walking distances of the targets can introduce significant noise into the detection results that propagate into tracking. Second, the number of people in the tracking area changes over time. Calculating the joint probability of an unknown number of targets is difficult and computationally expensive. To address these challenges, we adopt a well-known technique for target tracking in noisy environments: the probability hypothesis density (PHD) filter [13].

4.1 The Particle PHD Filter

The PHD is designed for tracking multiple targets and propagates only the first-order statistical moment, thus reducing the computational complexity. The PHD can be implemented efficiently with a particle filter, called a PHD filter. Next, we describe an *overview* and the main *operations* of the PHD, followed by the corresponding Particle Filter implementation.

Overview of the PHD. In essence, the PHD tracks multiple targets using a density function. The areas where the targets are located have a higher density (peaks) and the integral of the density function determines the number of targets. Formally, the general equation describing the PHD is given by,

$$\int_S D_{k|k}(\mathbf{x}|Z^{(k)})d\mathbf{x} = \hat{K}. \quad (5)$$

Given a set of binary detection outputs Z at timestep k (provided by the solar cells in our case), the function $D_{k|k}(\mathbf{x}|Z)$ represents the PHD for a tracking area $\mathbf{x}$ (an apartment in our case). For example, if the number of people in the tracking area is two, the integration of the PHD over that area should equal two [13, 12], which is represented by $\hat{K}$.

Particle Filter Implementation. The PHD can be implemented deploying a number of particles over the tracking area. The overview of the particle PHD implementation is shown in Fig. 10. To demonstrate the PHD filter algorithm, we divide the tracking area into three cells: cell 0 to cell 2 from left to right. Each cell is covered by a different solar sensor.

At the start of the algorithm, m number of particles with equal weight are generated. The particles are distributed uniformly over the entire tracking area ($\frac{m}{3}$ particles in each cell), as illustrated in stage 1 in Fig. 10. This initial distribution indicates that each tracking area is equally likely to have targets present. In Fig. 10, we use the sizes of the circles to indicate their weights. As the targets move, and the sensors provide information, the particles will increase their density on the cells where the targets are present. To estimate the number of targets, the integral in Eq. 5 can be transformed into the following sum,

$$\sum W_p(x) \approx \hat{K}, \quad (6)$$

where $W_p(x)$ represents the weight of the particles. Similar to the PHD, the summation of the weights of all particles will effectively provide an approximation of the number of targets in the area.

Operation 1: Prediction. Targets will move around, and they will also join or leave the tracking area. To predict these dynamics at each timestep, the value of the PHD filter is updated using the underlying motion models of the targets, which can be described with the following equation:

$$D_{k+1|k}(\mathbf{x}) = b_{k+1|k}(\mathbf{x}) + \int (p_s(\mathbf{w}) f_{k+1|k}(\mathbf{x}|\mathbf{w}) d\mathbf{w}. \quad (7)$$

Three types of dynamics are described in Eq. 7 to predict the location of a target at the next time step. *The survival probability* $p_s(\mathbf{w})$ describes the probability that the current targets remain in the tracking area. *The birth model* $b_{k+1|k}(\mathbf{x})$ describes the probability that a new target enters the tracking area, and *the* $f_{k+1|k}$ *models* describe the probability that a target moves to the neighboring cells.

Particle Filter Implementation. In our implementation, the survival probability is set to 0.7, the birth probability is set to 0.01, and the motion model follows the probabilities depicted at the bottom of stage 2 in Fig. 10. For demonstration clarity, only the particles initialized in cell 0 are shown starting from stage 2. To showcase how these three dynamics modify the density of particles, we use the transition from stage 2 to stage 3 in Fig. 10. A small amount of particles are deleted, while others are created (red dots). These two effects jointly represent the chance that targets enter or leave the tracking area. And based on the average walking speed of humans, the motion model predicts how far the particles can move.

Operation 2: Update. After predicting the movement of the targets, we use the solar sensing output Z_{k+1} received at time $k+1$ to update the weights of the particles. Particles where the sensor detects presence increase their weight and particles where the sensor indicates an absence of targets decrease their weight. The general equation describing this step is:

$$D_{k+1|k+1}(\mathbf{x}) \cong F_{k+1}(Z_{k+1}|\mathbf{x}) D_{k+1|k}(\mathbf{x}), \quad (8)$$

where the function $F_{k+1}(Z_{k+1}|\mathbf{x})$ indicates the update applied by the measurements.

Particle Filter Implementation. In the example, we have two targets present in cell 1 and cell 2, which are indicated by green shades in stage 4 in Fig. 10. The detection results of cell 1 and cell 2 are positive, therefore, the corresponding particles are updated with a higher weight than particles in cell 0. After the weights are updated, the expected number of targets can be calculated by summing up the weights of the particles.

The tracking algorithm provides the estimated positions for current targets at each timestep. A stream of estimated position data is received over time, which needs to be converted to tracks. We use a distance association to determine which track these estimations belong to. For each estimation at a time step, the distances between the new estimated position and the existing tracks are calculated and assigned into the closest tracks. This produces the final track of each target, which we will look at in the evaluation section.

4.2 Evaluation

To evaluate the performance of the tracking algorithm in a controlled manner we use a simulated setup. The layout of the scene and the paths are shown in Fig. 11. In the simulated scenario, two targets walk in opposite directions at the same speed. Target 2 enters the scene when Target 1 is in cell 1. The whole area is divided into five evenly spaced areas, each

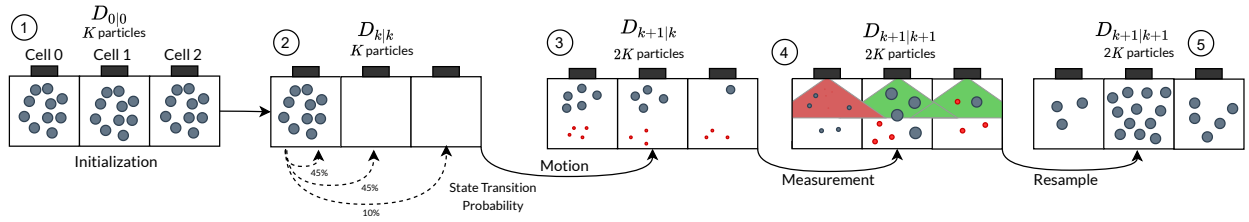


Figure 10: Overview of the particle filter implementation of the probability hypothesis density (PHD). The particles represent the probability density. When there is a cluster of highly-weighted particles in a certain position, it is likely that there is a target.

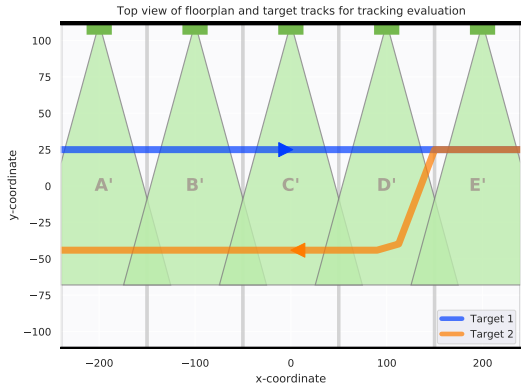


Figure 11: Top-view of floorplan used for evaluation of tracking algorithms.

with a solar cell. To test the resilience of the PHD filter to noisy data, we consider three types of scenarios:

- True detection (TD) - All sensors are able to correctly detect targets when they are in the corresponding detection areas.
- False negative (FN) - The sensor located in cell B' is set to always output 'False' and not detect any target.
- False positive (FP) - The sensor in cell B' is set to always output 'True' for 1 s when both targets are moving into cell D' ($2.75 \leq t \leq 3.75$).

Fig. 12 shows the simulated tracks and the estimated tracks from the particle PHD algorithm. In the test scenario TD, the algorithm is able to identify the tracks of the two targets correctly, as shown in Fig. 12a. For the scenario with FNs, the tracking algorithm struggles to correctly identify the track in cell B', as shown in 12b. This occurs because particles in cell B' will always receive lower weights, making it less likely to form a cluster. However, even though a target at cell B' cannot be detected, the tracking algorithm is able to successfully track both targets once they pass cell B'. This is because of two reasons. First, a few particles are able to move past the cell with consistent FNs according to the state transition model (note that the motion model in stage 2 has a 10% probability of moving two cells apart, Fig. 10). Second, thanks to the inherent birth model of PHD filters, a number of particles are generated in arbitrary areas at each timestep, allowing targets to re-emerge. The results of the test scenario with FPs are shown in Fig. 12c. While the presence of FPs have some effect on the estimated track, the algorithm is able to filter most of this noise because the motion model leaves only few particles in cell B'.

From the evaluation results, we can see that the tracking algorithm has the ability to ameliorate the presence of FNs and FPs that may not have been filtered by the detection framework in Section 3. This capability will be valuable in noisier environments, as presented next.

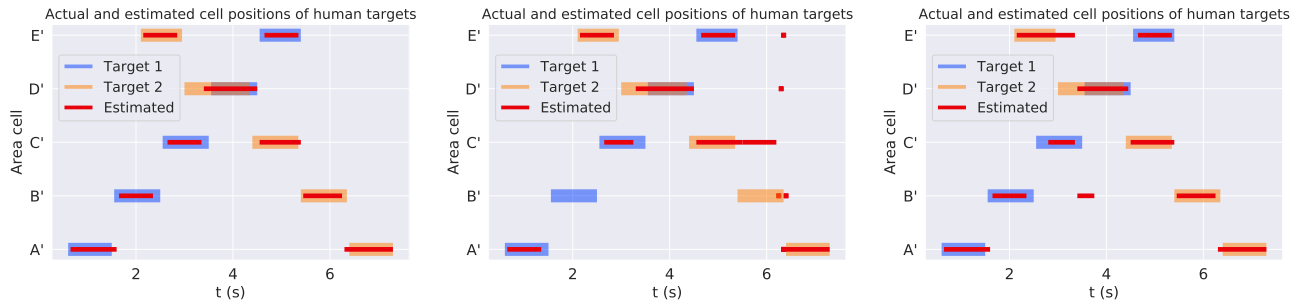
5 Experimental Evaluation

In this section, we describe the evaluation carried out to showcase the contribution of our work. We show that despite the challenges posed by the changing ambient light levels and the presence of multiple targets, the combination of a good placement of solar cells, detection using the BOCD algorithm and tracking using the PHD algorithm shows promise as a potential solution for indoor target tracking. The overview of the hardware setup is shown in Fig. 13a. Six sensing devices are deployed in the environment, and each sensing device consists of a solar cell, a microcontroller (MCU) and a battery. The sensing devices communicate with a base station, where the data processing, detection and tracking are done.

The tags designed for evaluation are built on an ESP32 microcontroller development board as shown in Figure 13b. A small 39 mm x 35 mm solar cell with a nominal output of 4 V, 35 mA is connected to the 10-bit ADC of the microcontroller. The circuit connecting the solar cell consists of a simple voltage divider to work within the input range of the ADC, and a 100 nF decoupling capacitor. The devices communicate the ADC data to the PC, which acted as the base station. Data is sent over Bluetooth at a rate of 100 Hz, which is more than necessary for running the algorithms but allows for extra analysis if needed. In our system, we use the ADC embedded in the microcontroller. There are low-power ADCs available in the market, for example, the ADS7042 consumes less than 1 µW at 1 kSPS, which can be used to further reduce the power consumption in sampling.

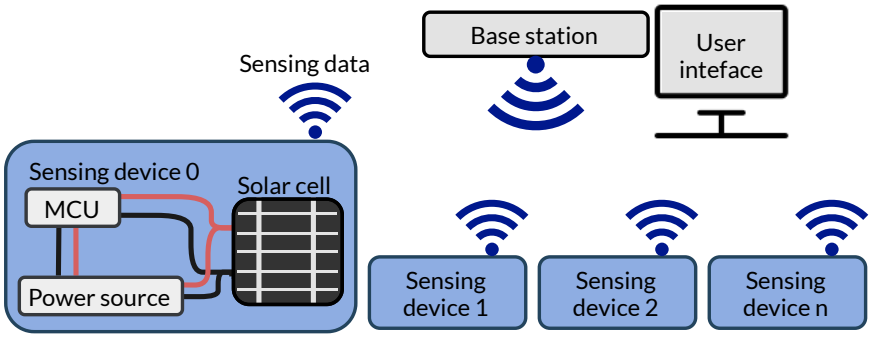
5.1 Experiment configurations

The experiments are carried out in an indoor residential environment. The floorplan of the environment is shown in Figure 14a, which also shows the positions and field-of-views of the solar cells. The area is divided into 6 smaller cell areas. Letters A-F represent the labels of the solar cells, while A'-F' represent the different areas covered by solar cells. Given that the distance from the paths to the sensors are similar to those considered in Section 2, we place the solar cells according to the optimal placement (35 cm high with a -25 angle), except for places where the furniture requires placing them higher, in which case we use a 'medium'

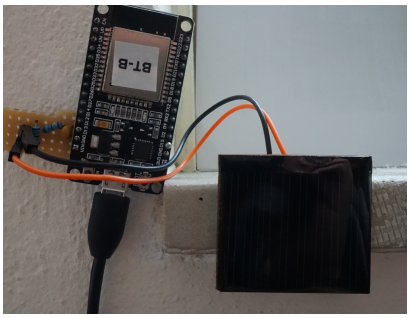


(a) 2 targets (b) 2 targets + FN (c) 2 targets + FP

Figure 12: Position estimation results for three different datasets.



(a) Overview of the hardware for the tracking system. The sensing devices communicate the data acquired from the solar cell to a base station over a wireless link.



(b) Microcontroller with solar cell attached, as used during the evaluation.

Figure 13: Hardware setup.

placement (110 cm high with a -10 angle).

The experiments have been carried out multiple times while changing three variables: the number of targets (1 or 2), the light source (only natural light during the day or artificial light from lamps at night) and the distance of the target to the sensors when passing by ('far' at 150 cm, or 'close' at 50 cm). Table 2 shows all the combinations of variables. During the experiment, each configuration is repeated three times. The targets are instructed to walk a fixed sequence of cells. They do this at a normal walking speed while also keeping to the requirement of passing the solar cells at 'far' or 'close' distance. The path trajectories are shown in Fig. 14b and Fig. 14c. The walking paths are:

- **1 target**
  - P1: (start outside B')-B'-C'-D' (wait approx. 5 seconds) -C'-B'-A'- (exit at cell A')
- **2 targets**
  - P1: (start outside B')-B'-C'-E'-F' (wait until P2 enters C') -E'-C'-B'-A' (exit at A')
  - P2: (start outside A', wait until P1 is at F') -A'-B'-C'-D' (wait until P1 enters B') -C'-B'(exit)

5.2 Results

The associated tracks are automatically produced by the algorithm. We use brackets to indicate False Positives. For example, the notation (A') indicates that the tracking output shows that the target passes by cell A', while in reality the target does not (caused by FPs). Similarly, square brackets indicate missing cells in the output. The notation [A'] indicates that the target passes by cell A', but this is missed by the tracking algorithm (caused by FNs). In Fig. 15, the α markers show the FPs in the detection results and their corresponding effects on the tracking results. The markers β show the FNs in the detection results and their corresponding effects on the tracking results.

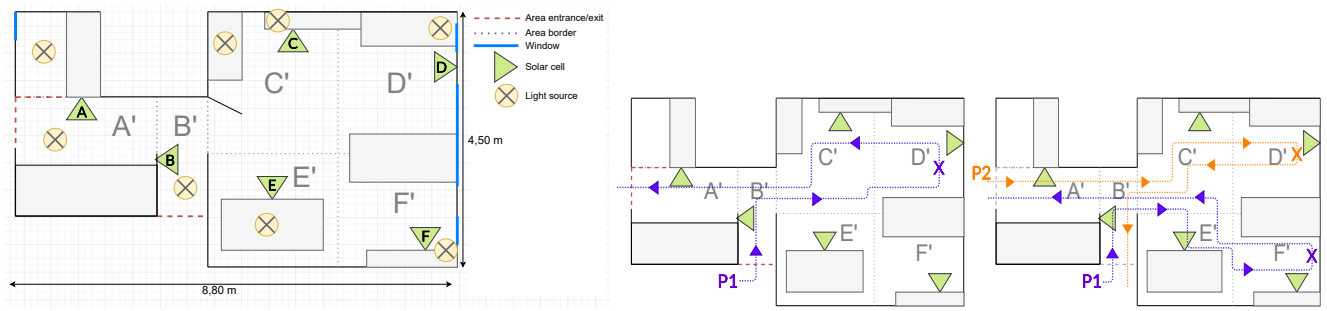
Table 2: All combinations of variables used for the experiment. Note that each combination was repeated three times. A far distance is 150 cm, a close distance is 50 cm.

No. of targets	Light	distance
1	Natural light	Close
1	Natural light	Far
1	Artificial light	Close
1	Artificial light	Far
2	Natural light	Close
2	Artificial light	Close

Among the six experiments shown in Table 2, we select three to showcase the performance of Inti. One benign setup, where the signals are relatively clean; one medium-level setup, which is noisier; and one hard setup, where there are several false positives and negatives. These three sample scenarios are described next.

5.2.1 An Easy Tracking Scenario

Fig. 15a shows an experimental situation with mostly correctly detected changepoints. This experiment was conducted with 1 target walking close to the solar cells during the day. The blue bars denote the outcome of the tracking al-

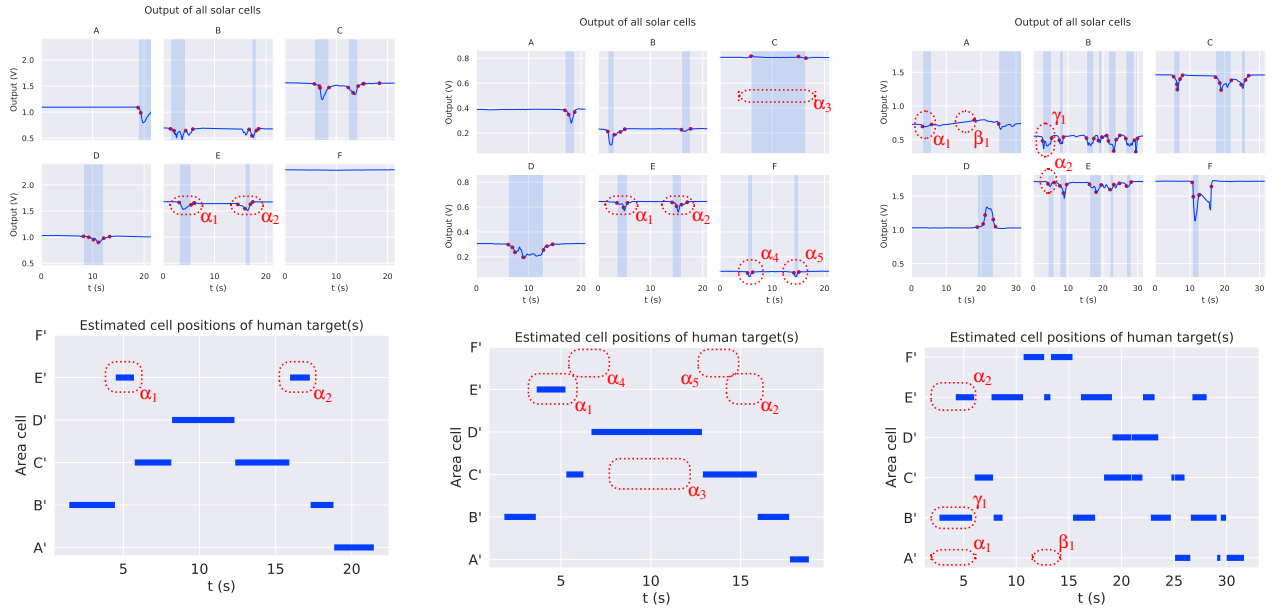


(a) Layout of the evaluation area.

(b) Walking path of one target.

(c) Walking paths of two targets.

Figure 14: Visualization of target tracks. Note that this only shows the sequence of cells visited. The actual position within the cell depends on whether the targets are instructed to pass the solar cells at 'far' or 'close' distance. An 'X' marks a location where the target stands still for a period of time.



(a) 1 target, natural light, close - straightforward tracking due to clear and correctly marked detections.

(b) 1 target, artificial light, far - tracking with some misdetections.

(c) 2 targets, natural light, close - tracking with many misdetections.

Figure 15: Solar cell outputs with marked detections and corresponding tracking outputs. Green indicates the actual track of target 1, and yellow indicates the actual track of target 2.

gorithm and the green bars the ground truth. In this scenario, as the target walks close to the solar cells, clear changepoints can be detected, resulting in correct detections, except for the FPs marked with α_1 and α_2 in cell E'. These FPs also appear in the tracking outputs, but they are not significant errors because they occur due to the presence of contiguous cells in the layout of the apartment. Note that as the target walks from cell B' to cell C', the path is at the border of cell E', which also detects the presence of the target. The tracking output from the algorithm is B'-(E')-C'-D'-C'-(E')-B'-A'. In a way, this 'contiguous' redundancy validates the robustness of the approach.

5.2.2 *A Medium Tracking Scenario*

Fig. 15b shows an instance where more misdetections are present. This experiment was conducted with 1 target walking far from the solar cells at night. The walking path of the target is the same as in Fig. 15a. As the target walks far from the solar cell, the variation of light on the tag is smaller, leading to noisier signals. In this scenario, the detection algorithm recognizes when the target appears in the area through cell B', but assumes that the target remains for a long time in cell C', which is incorrect. This is indicated by α_3 in Fig. 15b. This occurs because, in cell C', the system merges two short detection events as one long detection: the signals ending the first event and starting the second are too weak to be detected (FNs), and hence the system combined the starting changepoint of the first event with the ending changepoint of the second event. In addition, when the target walks far from the solar cells, unintended disturbances can occur to cells that they do not pass through, causing FPs. This is indicated by α_4 and α_5 in cell F'. For α_1 and α_2, the cause of these FPs is similar to the previous case in Fig. 15a, as the target follows the same path. Note however that in spite of the noisier data, four out of the five misdetections

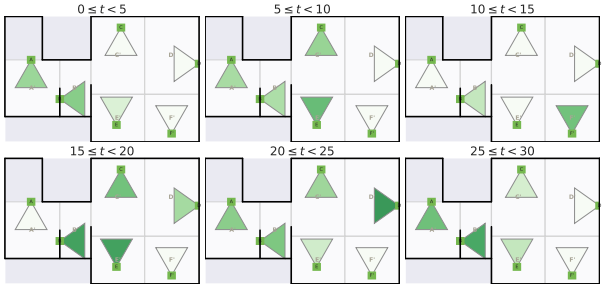


Figure 16: Visual representation of the detection output for all solar cells for different time windows. A darker color means that the solar cell is detecting a target for a longer time period during that window.

are filtered out correctly, and the only FP remaining in the estimated blue track is α_2, which as stated before is not a major issue because it is due to contiguous cells. The tracking output from the algorithm is B'-(E')-C'-D'-C'-B'-A'.

5.2.3 *A Difficult Tracking Scenario*

Fig. 15c shows the performance of an experiment conducted with 2 targets walking close to the solar cells during the day. To interpret Fig. 15c better, Fig. 16 shows the detection output for different time windows. As this experiment was conducted during the day, the results are subjected to the inconsistency of the natural light levels. Natural light is stronger than artificial light and has a more horizontal radiation. As can be seen from Fig. 15c, many sudden events happen, causing the natural light level to fluctuate, which results in noisy detection outputs.

In the first 5 seconds, target P1 enters the area from cell B', which is detected correctly, as marked with γ_1. However, solar cells A and E have FPs at the same time, which are marked with α_1 and α_2. In this case, the tracking algorithm is able to filter out those FPs and correctly estimate the position to be in cell B'. However, it also mistakenly identifies that the target passes cell E', marked by α_2. Similar to the previous scenarios, as cell E' borders with most cells, it has many FPs. Between $t = 10s$ and $t = 15s$, the target in cell A' is not recognized, which is marked by β_1 in Fig. 15c.

Overall, similar to the prior scenarios, despite the FPs and FNs of the detection results, the tracking algorithm is still able to recognize the majority of the target paths as shown by the overlap between the estimated paths (blue bars) and the ground truth (green and yellow bars).

In conclusion, being able to correctly track passing targets in an indoor environment is a joint effort between solar cell placements, the detection algorithm and the tracking algorithm. In a real indoor deployment, the solar cell outputs tend to be quite noisy and are subject to many FNs and FPs. This can cause misdetections to occur. However, the tracking algorithm is able to filter out much of the noise.

5.3 Discussion

This work provides a starting point towards realizing energy-neutral indoor tracking with solar cells. In this section, we discuss the limitations of the current system that can be improved on.

1. The inputs to the simulator are based on assumptions of the environment, such as the type of reflections and the reflection coefficients from walls and furniture based on their materials. An experimental characterization can be carried out to obtain more accurate reflection characteristics, however, they will result in a more time-consuming process. In addition, only artificial light is considered in the simulator. It would be interesting to include the effects of natural light.
2. A limitation of the system is that the prototype developed for evaluation requires an external power source. A more detailed power and circuitry analysis is required to power the system solely with the solar cells. The research community is working on optimal methods to attain simultaneous harvesting and sensing, and such methods can be integrated with our solution. The value of our work is to showcase the challenges and potential solutions related to using solar cells for detection and tracking.

6 Related Work

6.1 Visible Light Sensing

Visible light sensing (VLS) focuses on sensing objects by exploiting the influence of the object on visual light within the environment [4]. Visible light positioning (VLP) is a type of VLS concerned with finding the positions of targets in an area. In VLP with passive users, the system detects the presence of targets based on the shadows and the reflections they cast on a light sensor.

Many methods in passive VLP require the use of *active sources*. An active source is a light source that provides illumination, but has data modulated in the light as well by an adapted LED driver [7, 9, 16, 24]. A few studies have explored the use of passive sources, for which no modifications to the existing lighting infrastructure have to be made. However, these studies require the use of offline learning, which is called *fingerprinting*, prior to deployment. During the offline learning, the outputs of different receiver signal strengths are stored for different situations. Once deployed, it uses the knowledge of these values to estimate the state of the current situation [8, 5, 22]. A disadvantage of fingerprinting is that it requires a separate learning phase to implement for each new environment. Furthermore, it is not very robust in situations where the lighting situation is not constant as the signal strength is not only influenced by the presence of targets, but by environmental lighting conditions as well.

Similar to some of the above studies, our approach does not require users to carry any device. The main advantage of our approach, however, is that we do not require modifications of the lighting infrastructure or heavy fingerprinting phases.

6.2 Sensing with Solar Cells

Many types of sensors can be used for indoor tracking, such as cameras, infrared or ultrasound sensors [3, 19, 6]. While modifications can be applied to reduce the resolution such that these sensors do not have sufficient resolution to identify people, solar cells are one step further in preserving privacy, since they are essentially a single-pixel device and we are treating the output as a binary data point (presence/absence). Another main advantage of employing solar cells

as sensors is the potential of providing simultaneous power generation and sensing. In mass-deployed systems, this reduces the need for battery replacements. There is a growing interest in adopting solar cells for indoor applications. However, most of these designs focus on active communication and interaction applications with the solar cells, and very few examples utilize them for sensing. For example, in SolarGest [11], a transparent solar panel is used to distinguish a number of hand gestures, but the range is limited to only a few centimeters. In [21] a solar cell is combined with a comparator circuit to generate a binary output without the use of an ADC. This light-based approach combined with RF backscatter results in a battery-free device. However, the basic thresholding circuit used for generating the binary output is only suitable for a very controlled environment. In [18], solar cells are used as wearable devices for tracking human activities. However, this type of tracking requires the active participation from users, as users have to attach solar cells to their bodies. Our work is motivated by these studies about the use of solar cells as sensors, but we target a challenging application with longer distances, and noisier setups, between the solar cells and the targets.

7 Conclusion

In Inti, we propose a novel multi-target indoor tracking system based on solar cells. Our framework proposes three components. First, we develop a ray-tracing simulator to determine the best placements for solar cells. Our approach leads to placements that result in a high signal to noise ratio as targets walk by. Second, we apply an online Bayesian changepoint detection method to the received solar cell signals to convert them into binary detection signals. The Bayesian method is able to achieve consistent detection results in different illumination conditions. Third, we apply the PHD filter to combine the detection results from different tags. This approach allows us to filter several false positives/negatives and to perform tracking without knowing the actual number of targets in the environment. Finally, we evaluate our framework in a small apartment, and show that Inti is able to track people under different lighting conditions, both during the day with uncontrolled lighting, as well as at night with controlled lighting.

8 Acknowledgments

The authors would like to thank the reviewers and shepherd, Rui Tan, for their feedback. This work has been funded by the European Union's H2020 programme under the Marie Skłodowska Curie grant agreement ENLIGHTEM No. 814215, and by the Dutch Research Council (NWO) with a TOP-Grant with project number 612.001.854.

9 References

[1] R. P. Adams and D. J. C. MacKay. Bayesian Online Changepoint Detection. oct 2007.

[2] J. Clemente, F. Li, M. Valero, and W. Song. Smart seismic sensing for indoor fall detection, location, and notification. *IEEE Journal of Biomedical and Health Informatics*, 24(2):524–532, 2020.

[3] A. S. Crandall and D. J. Cook. Coping with multiple residents in a smart environment. *Journal of Ambient Intelligence and Smart Environments*, 1(4):323–334, 2009.

[4] K. Deprez, S. Bastiaens, L. Martens, W. Joseph, and D. Plets. Passive visible light detection of humans. *Sensors (Switzerland)*, 20(7), 2020.

[5] N. Faulkner, F. Alam, M. Legg, and S. Demidenko. Watchers on the Wall: Passive Visible Light-Based Positioning and Tracking with Embedded Light-Sensors on the Wall. *IEEE Transactions on Instrumentation and Measurement*, 69(5):2522–2532, 2020.

[6] T. W. Hnat, E. Griffiths, R. Dawson, and K. Whitehouse. Doorjamb: Unobtrusive room-level tracking of people in homes using doorway sensors. In *SenSys 2012 - Proceedings of the 10th ACM Conference on Embedded Networked Sensor Systems*, 2012.

[7] M. Ibrahim, V. Nguyen, S. Rupavatharam, M. Jawahar, M. Gruteser, and R. Howard. Visible light based activity sensing using ceiling photosensors. *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, 03-07-Octo:43–48, 2016.

[8] D. Konings, N. Faulkner, F. Alam, E. M. Lai, and S. Demidenko. FieldLight: Device-Free Indoor Human Localization Using Passive Visible Light Positioning and Artificial Potential Fields. *IEEE Sensors Journal*, 20(2):1054–1066, 2020.

[9] T. Li, Q. Liu, and X. Zhou. Practical Human Sensing in the Light. *GetMobile: Mobile Computing and Communications*, 20(4):28–33, 2017.

[10] Y. Li, T. Li, R. A. Patel, X. D. Yang, X. Zhou, and C. P. Authors. Self-powered gesture recognition with ambient light. *UIST 2018 - Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*, pages 595–608, 2018.

[11] D. Ma, W. Hu, G. Lan, M. B. Upama, M. Youssef, M. Hassan, and A. Uddin. Solargest: Ubiquitous and Battery-free Gesture Recognition using Solar Cells. *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, 2019.

[12] R. Mahler. PHD filters for nonstandard targets, I: Extended targets. *2009 12th International Conference on Information Fusion, FUSION 2009*, pages 915–921, 2009.

[13] R. P. Mahler. Multitarget Bayes Filtering via First-Order Multitarget Moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1152–1178, 2003.

[14] K. P. Murphy. Conjugate Bayesian Analysis of the Gaussian Distribution. *Def*, 1(7):1–29, 2007.

[15] S. Nannuru, Y. Li, Y. Zeng, M. Coates, and B. Yang. Radio-frequency tomography for passive indoor multitarget tracking. *IEEE Transactions on Mobile Computing*, 12(12):2322–2333, 2013.

[16] V. Nguyen, M. Ibrahim, S. Rupavatharam, M. Jawahar, M. Gruteser, and R. Howard. Eyelight: Light-and-Shadow-Based Occupancy Estimation and Room Activity Recognition. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pages 351–359. IEEE, apr 2018.

[17] J. Peddie. Applications of Ray Tracing. In *Ray Tracing: A Tool for All*, pages 91–128. Springer International Publishing, Cham, 2019.

[18] M. M. Sandhu, S. Khalifa, K. Geissdoerfer, R. Jurdak, and M. Portmann. Solar: Energy positive human activity recognition using solar cells. In *2021 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 1–10, 2021.

[19] S. Tao, M. Kudo, B. N. Pei, H. Nonaka, and J. Toyama. Multiperson Locating and Their Soft Tracking in a Binary Infrared Sensor Network. *IEEE Transactions on Human-Machine Systems*, 45(5):550–561, 2015.

[20] N. Tatbul, T. J. Lee, S. Zdonik, M. Alam, and J. Gottschlich. Precision and recall for time series. *Advances in Neural Information Processing Systems*, 2018-Decem(NeurIPS):1920–1930, 2018.

[21] A. Varshney, A. Soleiman, L. Mottola, and T. Voigt. Battery-free visible light sensing. *VLCS 2017 - Proceedings of the 4th ACM Workshop on Visible Light Communication Systems, co-located with MobiCom 2017*, pages 3–8, 2017.

[22] Y. Yang, J. Hao, J. Luo, and S. J. Pan. CeilingSee: Device-free occupancy inference through lighting infrastructure based LED sensing. *2017 IEEE International Conference on Pervasive Computing and Communications, PerCom 2017*, pages 247–256, 2017.

[23] M. Yasir, S. W. Ho, and B. N. Vellambi. Indoor position tracking using multiple optical receivers. *Journal of Lightwave Technology*, 34(4):1166–1176, 2016.

[24] S. Zhang, K. Liu, Y. Ma, X. Huang, X. Gong, and Y. Zhang. An Accurate Geometrical Multi-Target Device-Free Localization Method Using Light Sensors. *IEEE Sensors Journal*, 18(18):7619–7632, 2018.