Dynamic Occupancy Grid Filtering for Multi-target Tracking and Goal Location Inference

Haruki Nishimura – 06045657
AA273 State Estimation and Filtering for Aerospace Systems
Prof. Mac Schwager

June 16, 2017

Abstract

In this paper, we propose a framework for multi-target tracking that simultaneously estimates unknown goal locations for dynamic obstacles (i.e. pedestrians). Our method is based on an occupancy grid representation of the environment and a spatial independence assumption to the grid cells. We employ an optimal decision making framework to model behavioral intentions of pedestrians, and use a set of offline policies to estimate pedestrian intentions in the filtering process. We demonstrate through simulation that our algorithm can identify static obstacles, track pedestrians and estimate different pedestrian intentions without any data association required.

1 Introduction and Related Work

In the field of mobile robotics, a crucial capability for an intelligent agent is to perceive and model the surrounding environment so that the agent can take appropriate actions to achieve its goal. This perception step still remains as a challenging task by itself, since in many cases the sensory system can only obtain imperfect information of the world. For example, sensor measurements are often corrupted by noise, and sometimes a significant portion of the area of interest is occluded. The issue of imperfect sensing becomes critical especially when the agent needs to operate carefully alongside densely populated static and dynamic obstacles while avoiding collisions.

Probabilistic methods are preferred in such a case because they can naturally incorporate sensing uncertainty into the model. A traditional way of representing moving obstacles is to explicitly model each of them and track their positions through object-based multi-target tracking methods. Although this type of algorithms is often used in computer vision systems, it is not robust to complete object occlusions in general, and the data association remains as a post-processing step [1]. An alternative approach is based on occupancy grid representations, in which the surrounding environment is discretized into a finite number of cells and the spatial occupancy of them are tracked [2]. This approach is more advantageous than the object-based methods in that it is more robust to object occlusions, and also the data association step can be avoided if it is not required; if the agent is only tasked with recognizing moving obstacles for short-term collision awareness, for example, the positions and the velocities of the obstacles constitute sufficient information. Furthermore, sensor fusion from heterogeneous data sources can be easily managed with spatial occupancy representations [3].

The Bayesian Occupancy Filter [4] is a generic filtering framework where the motion of the cell occupancies is inferred in the Bayesian state estimation scheme. The joint state consists of the occupancy state for each cell and the associated velocity distribution. The Bayesian Occupancy Filter assumes that the cells are spatially independent, which makes the computation tractable and highly parallelizable in practice. The filter in its original form tracks both the static and dynamic obstacles using the discrete Bayesian filter, where the velocity distribution associated with each cell is also discretized into grids. This complete discrete representation results in a four dimensional joint state where most of those 4D-cells are not occupied, and is also confronted with aliasing issues when the frame of reference is rotated and translated with the sensory system. Nègre et al. present the Hybrid Sampling Bayesian Occupancy Filter [5] to address these problems. The main idea is to model dynamic occupancies with a set of moving particles to allow for continuous position and velocity distributions, while static occupancies are still tracked by the discrete Bayesian filter. The authors claim that their approach improves accuracy in terms of the velocity estimation with a reduced amount of data to represent the joint belief state. The same authors also propose a variant of the Hybrid Bayesian Occupancy Filter, called the Conditional Monte Carlo Dense Occupancy Tracker [6]. In this new formulation the state of each cell is defined separately from the cell occupancy, and the occupancy is indirectly inferred from the cell state as a consequence.

While the Bayesian Occupancy Filter and its variants are suitable for a low-level perception system for mobile robots and intelligent vehicles, they are not immediately applicable to an accurate prediction of the future state, which is critical to long-term collision avoidance and safe motion planning [7]. This is because most of the Bayesian Occupancy Filters assume a simple motion model for dynamic obstacles, such as a constant velocity model with white Gaussian acceleration noise. One of the few exceptions is the Bayesian Occupancy Filter Using Map [8, 9] developed by Gindele et al., where they incorporate prior knowledge about motion preferences of dynamic obstacles hypothesized from map data in order to improve the accuracy of predictions. However, since prior map knowledge itself only gives the geographic information of the environment and does not validate a complete motion model for the dynamic obstacles in it, the quality of predictions is still questionable. In addition, the environment needs to be highly structured for this method to work.

A more direct approach to improving the prediction accuracy is to explicitly model behavioral intentions of moving obstacles and estimate them along with the positions and velocities through the Bayesian filtering. The future trajectories can be then predicted by propagating in time the intention-dependent motion model. If the moving obstacles are pedestrians, a common way of modeling their behavioral intentions is to estimate their goal locations. Best and Fitch [10] formalize the problem of goal location inference for a single moving target as a recursive Bayesian state estimation problem to demonstrate the improved accuracy of long-term trajectory predictions. Bai et al. [11] employ the pedestrian goal location inference to safely operate an autonomous golf cart near humans. Although both methods assume no uncertainty in position and velocity information, the idea of filtering behavioral intentions seems particularly effective for predicting future trajectories of dynamic obstacles, leading eventually to safe motion planning.

With the discussion above, we present a framework for multi-target tracking that also esimates unknown goal locations for dynamic obstacles (i.e. pedestrians) simultaneously. Our method is based on the occupancy grid representation of the environment. More specifically we extend the Conditional Monte Carlo Dense Occupancy Tracker [6] by integrating into it a behavioral intention model for the pedestrians. We employ an optimal decision making framework to model and estimate the pedestrian intentions. The approach proposed in this paper does not require an explicit data association process to track different pedestrian intentions, since the notion of pedestrian intentions is a property that belongs to cells, not to the pedestrians themselves.

The rest of the paper is organized as follows. The pedestrian motion model is defined in Section 2. The filtering framework is formalized in Section 3. Section 4 demonstrates the simulation results and Section 5 concludes this paper.

2 Pedestrian Motion Model

We assume that the configuration space $\mathcal{W} \subset \mathbb{R}^2$ is closed and discretized into regular grids. The resolution of the discretization is set to be the same as the occupancy sensor resolution for simplicity. Each cell represents a possible position $(p_x, p_y)^{\mathrm{T}}$ for one or more pedestrians moving in $\mathcal{W}$.

The configuration space $\mathcal{W}$ is populated with static obstacles $\mathcal{O}$, and there exists a set of possible goal locations $\mathcal{G} \subset \mathcal{W} \backslash \mathcal{O}$ for the pedestrians. The static obstacle set $\mathcal{O}$ and the goal location set $\mathcal{G}$ are known to the sensory system as prior map knowledge, but a unique goal location $g^a \in \mathcal{G}$ associated with each individual pedestrian a remains unknown.

2.1 Generic motion model

In modeling the pedestrian motion, we simply ignore any interactions between pedestrians. The pedestrian-pedestrian interactions could be left for future work. Consequently, each individual pedestrian a follows the same generic discrete motion model presented below.

$$p^a_{x,t+1} = p^a_{x,t} + v^a_{x,t} + w^a_{px,t} \tag{1}$$

$$p^a_{y,t+1} = p^a_{y,t} + v^a_{y,t} + w^a_{py,t} \tag{2}$$

$$v^a_{x,t+1} = v^a_{x,t} + u^a_{x,t} + w^a_{vx,t} \tag{3}$$

$$v^a_{y,t+1} = v^a_{y,t} + u^a_{y,t} + w^a_{vy,t} \tag{4}$$

The equations (1) and (2) describe the position transitions in x and y directions from time t to $t+1$, and similarly (3) and (4) are for the velocity transitions. The velocity is bounded by the maximum speed $\max |v|$ to prevent pedestrians from moving too fast. This double integrator model is controlled by the discrete acceleration input $(u^a_{x,t}, u^a_{y,t})^{\mathrm{T}}$ to the velocity and is subject to discrete process noise $(w^a_{px,t}, w^a_{py,t}, w^a_{vx,t}, w^a_{vy,t})^{\mathrm{T}}$. Although the model defined above is holonomic, the method proposed in this paper can be applied to motion models of non-holonomic dynamic obstacles.

2.2 Intention modeling: Markov decision process

In order for the sensory system to obtain a clue for estimating the unknown goal locations, we need to provide it with the intention-dependent pedestrian motion model. Let x^a_t denote the joint state $(p^a_{x,t}, p^a_{y,t}, v^a_{x,t}, v^a_{y,t})^{\mathrm{T}}$ for pedestrian a. The intention-dependent motion model is the state transition model that is conditioned on the unknown goal location;

$$P(x^a_{t+1} \mid x^a_t, g^a). \tag{5}$$

If real data collected and labeled offline are available, the intention-dependent motion model above could be learnt from them. It is also possible to employ a heuristic to define (5) as discussed in [10, 11]. Nevertheless, in this work we take a different approach to simulate pedestrian motion; we define a sensible reward function for each goal configuration, and compute a policy assuming that the pedestrian acts optimally under that reward function and the generic discrete motion model (1) - (4). We seek for the optimal policy by formulating the problem as a Markov Decision Process (MDP) [12]. Explained below is a brief overview of the Markov Decision Process and how to find (5) from the resulting optimal policy.

2.2.1 MDP overview

A Markov Decision Process (MDP) consists of a state space $\mathcal{X}$, an action space $\mathcal{U}$, a state transition function $T(x^a_{t+1} \mid x^a_t, u^a_t) = P(x^a_{t+1} \mid x^a_t, u^a_t)$, and a reward function $R(x^a_t, u^a_t)$. In the case of infinite horizon MDPs, a discount factor $\gamma < 1$ is also introduced. A policy is a function that maps the past history of states and actions to a new action, which is reduced to a function of the current state under the Markov assumption. An optimal policy $\pi^* : \mathcal{X} \rightarrow \mathcal{U}$ is one that maximizes the expected cumulative reward or value U associated with each state in $\mathcal{X}$. In general there can be multiple optimal policies, but the optimal value is unique. The value of an optimal policy satisfies the Bellman equation;

$$U^*(x_t^a) = \max_{u_t^a} \left(R(x_t^a, u_t^a) + \gamma \sum_{x_{t+1}^a} T(x_{t+1}^a \mid x_t^a, u_t^a) U^*(x_{t+1}^a) \right). \tag{6}$$

If the state space $\mathcal{X}$ is discrete and relatively small, we can use dynamic programming methods to solve for an exact optimal policy. The value iteration is one of such algorithms, which initializes U to a bounded value (typically zero) and iteratively updates the value function by performing the following Bellman update for all the states $x^a \in \mathcal{X}$ synchronously until convergence.

$$U(x^a) \longleftarrow \max_{u^a} \left(R(x^a, u^a) + \gamma \sum_{x^{'a}} T(x^{'a} \mid x^a, u^a) U(x^{'a}) \right) \tag{7}$$

The optimal policy can be then extracted from the optimal value.

$$\pi^*(x^a) = \arg\max_{u^a} \left(R(x^a, u^a) + \gamma \sum_{x^{'a}} T(x^{'a} \mid x^a, u^a) U^*(x^{'a}) \right) \tag{8}$$

2.2.2 MDP for intention modeling

The reward function for our Markov Decision Process consists of four additive terms that prescribe a sensible pedestrian behavior.

$$R(x^a, u^a) = R_{obs}(x^a) + R_{goal}(x^a) + R_u(u^a) + R_v(x^a) \tag{9}$$

- If the pedestrian is in a state that results in a collision into a static obstacle $\mathcal{O}$ at the next time step, there is a large negative reward $R_{obs}(x^a)$.
- When the pedestrian arrives at its own goal location $g^a \in \mathcal{G}$, a large positive reward $R_{goal}(x^a)$ is given.
- There is a small negative reward $R_u(u^a)$ for acceleration and deceleration. This is proportional to the amount of absolute acceleration and encourages smooth motion.
- A small positive reward $R_v(x^a)$ is proportional to the absolute speed of the pedestrian. This is to encourage faster motion to get to the goal quickly.

Since there are $|\mathcal{G}|$ possible different goal locations, there are $|\mathcal{G}|$ different reward functions, one for each goal configuration. Once these reward functions are defined, we can compute the corresponding $|\mathcal{G}|$ optimal policies using the value iteration (7) as long as the state space is discrete and not too large. Let

$R_{g^a}(x^a, u^a)$ be the reward function with the goal location being $g^a \in \mathcal{G}$. The value iteration computes the optimal policy $\pi^*_{g^a}(x^a)$. Finally, the intention-dependent motion model is given by the following closed-loop state transition distribution.

$$P(x^a_{t+1} \mid x^a_t, g^a) \triangleq P\left(x^a_{t+1} \mid x^a_t, \pi^*_{g^a}(x^a_t)\right) \tag{10}$$

In practice, we only compute offline and store the set of optimal policies $\{\pi^*_g \mid g \in \mathcal{G}\}$ instead of computing (10), in order to save disk space. The intention-dependent motion model can be obtained from π^*_g and the generic motion model (1) - (4) upon filtering.

3 Dynamic Occupancy Grid Filtering

The filtering method presented here is based on the Hybrid Sampling Bayesian Occupancy Filter [5] and the Conditional Monte Carlo Dense Occupancy Tracker [6]. The notation defined below is mostly inherited from the previous work.

- C : Index of a cell that represents the cell position (p_x, p_y).
- C^{-1}: Index that identifies an antecedent of the occupancy in cell C.
- S : Occupancy state of the cell C at the current time.
- S^{-1}: Occupancy state of the antecedent cell at the previous time.
- V : Occupancy velocity of the cell C at the current time.
- V^{-1}: Occupancy velocity of the antecedent cell at the previous time.
- Z : Sensor measurement.

A key difference of the proposed approach from the previous work is that the cell occupancy state S explicitly represents the dynamic obstacles acting under different intentions.

$$S \in \{s, d_1, d_2, \dots, d_{|\mathcal{G}|}, e\}. \tag{11}$$

In (11), $S = s$ indicates that the cell C is occupied by a static obstacle. On the other hand, $S = d_i$ means that the cell is occupied by a dynamic obstacle whose goal location index is $i \in \{1, \dots, |\mathcal{G}|\}$. $S = e$ means that the cell is empty.

3.1 Joint distribution

As is the case with most of the occupancy grid filtering methods, we assume that the cell states are spatially independent of each other at each time step. Consequently, the joint state of the entire configuration space $\mathcal{W}$ can be decoupled into individual cell states. The variable dependency structure associated with the specific cell C is illustrated in Figure 1.

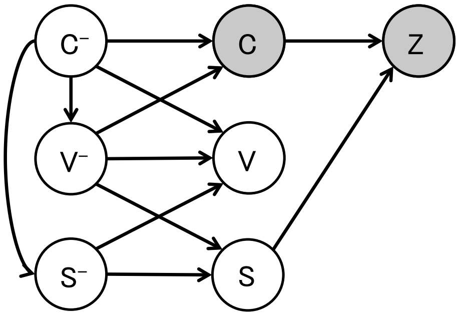


Figure 1: Bayesian network representing the variable dependency structure associated with cell C. Shaded nodes are evidence variables.

$$\begin{aligned} P(C, C^1, S, S^{-1}, V, V^{-1}, Z) =& P(Z \mid S, C) P(S \mid S^{-1}, V^{-1}) P(C \mid V^{-1}, C^{-1}) \\ & P(V \mid V^{-1}, S^{-1}, C^{-1}) P(S^{-1}, V^{-1} \mid C^{-1}) \\ & P(C^{-1}) \end{aligned} \tag{12}$$

Each conditional distribution in the factored joint distribution (12) can be interpreted as follows.

- $P(Z \mid S, C)$ is the measurement model for cell C. The possible value for measurement Z is *occ* or *nocc*, indicating whether the cell C is occupied by any type of obtacles or not. The measurement noise such as false alarms and missed detections can be specified here.
- $P(S \mid S^{-1}, V^{-1})$ is the occupancy state transition model, which can be represented by an occupancy state transition matrix. An example of the transition matrix is presented below. The empty state is less durable than the static or the dynamic parts.

	$S^{-1} = s$	$S^{-1} = d_1$	$\dots$	$S^{-1} = d_{\lvert\mathcal{G}\rvert}$	$S^{-1} = e$
$S = s$	0.9999	$f(\lvert v^{-1}\rvert)$	$\dots$	$f(\lvert v^{-1}\rvert)$	$0.001/(1+\lvert\mathcal{G}\rvert)$
$S = d_1$	$0.0001/\lvert\mathcal{G}\rvert$	$1 - f(\lvert v^{-1}\rvert)$	$\dots$	0	$0.001/(1+\lvert\mathcal{G}\rvert)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$S = d_{\lvert\mathcal{G}\rvert}$	$0.0001/\lvert\mathcal{G}\rvert$	0	$\dots$	$1 - f(\lvert v^{-1}\rvert)$	$0.001/(1+\lvert\mathcal{G}\rvert)$
$S = e$	0	0	$\dots$	0	0.999

Table 1: An example of the occupancy state transition matrix.

$f(|v^{-1}|)$ is a function of the previous speed $|v^{-1}|$ that changes a slow or stopped dynamic obstacle to a static obstacle. f can be any monotinically decreasing function that satisfies $f = 1$ when $|v^{-1}| = 0$ and $f \approx 0$ when $|v^{-1}|$ is sufficiently high, such as a decaying exponential.

- $P(C \mid V^{-1}, C^{-1})$ prescribes the cell reachability. It determines the probability that cell C is reached by the occupancy in the antecedent cell C^{-1} with the previous velocity V^{-1}. For static and empty occupancies, V^{-1} is always 0 and $C = C^{-1}$ in a deterministic sense.
- $P(V \mid V^{-1}, S^{-1}, C^{-1})$ is the velocity transition distribution. If the previous occupancy state S^{-1} is s (static) or e (empty), then the new velocity V remains 0 as they do not move. Otherwise this distribution is the same as the intention-dependent motion model for pedestrians (5)(10), which can be found from the offline policies $\{\pi_g^* \mid g \in \mathcal{G}\}$ and the generic velocity transition model (3)(4).
- $P(S^{-1}, V^{-1} \mid C^{-1})$ is the distribution over the occupancy state and the velocity for cell C^{-1} at the previous time step. This distribution is updated at each time step for all the cells.
- Lastly, $P(C^{-1})$ defines the prior distribution over all possible antecedents of cell C. A uniform distribution is chosen because the cell is considered to be equally reachable from any cells with no additional information.

3.2 Posterior computation

With the conditional distributions defined above, we can perform the Bayesian filtering to find the posterior distribution $P(S, V \mid Z, C)$ from the prior $P(S^{-1}, V^{-1} \mid C^{-1})$.

The predict step is given by

$$P(S, V \mid C) = \frac{\sum_{C^{-1}S^{-1}V^{-1}} P(C, C^{-1}, S, S^{-1}, V, V^{-1})}{\sum_{C^{-1}SS^{-1}VV^{-1}} P(C, C^{-1}, S, S^{-1}, V, V^{-1})}, \tag{13}$$

where

$$\begin{aligned} P(C, C^{-1}, S, S^{-1}, V, V^{-1}) = & P(S \mid S^{-1}, V^{-1}) P(C \mid V^{-1}, C^{-1}) \\ & P(V \mid V^{-1}, S^{-1}, C^{-1}) P(S^{-1}, V^{-1} \mid C^{-1}) P(C^{-1}). \end{aligned} \tag{14}$$

The update step is given by

$$P(S, V \mid Z, C) = \frac{P(Z \mid C, S) P(S, V \mid C)}{\sum_{SV} P(Z \mid C, S) P(S, V \mid C)}. \tag{15}$$

3.3 Practical Implementation

One advantage to the Hybrid Sampling Bayesian Occupancy Filter [5] and the Conditional Monte Carlo Dense Occupancy Tracker [6] is that they realize a more compact representation in terms of the number of required parameters than other dynamic occupancy grid filtering methods. This data reduction is achieved by the use of particle filtering to track dynamic obstacles. In this work, the same framework is employed to implement the filter; a set of particles $\{\chi_i\}$ represents the dynamics of the moving areas, while the static part ($S = s$ and $S = e$) is managed by the discrete Bayesian filter (13) - (15).

A particle χ_i is composed of a position $(p^i_x, p^i_y)^{\mathrm{T}}$, a velocity $(v^i_x, v^i_y)^{\mathrm{T}}$, a dynamic occupancy state $d^i \in \{d_1, \ldots, d_{|\mathcal{G}|}\}$, the corresponding goal location $g^i \in \mathcal{G}$, and a particle weight w_i. Both the position and the velocity vectors are in $\mathbb{R}^2$. This is to estimate the velocity more accurately in general[1], and also to avoid the aliasing issues.

3.3.1 Predict step for particles

In the predict step, the particles are moved according to the intention-dependent motion model.

$$\begin{aligned} u^i_t &= \pi^*_{g^i}(p^{proj,i}_t, v^{proj,i}_t) && (16)\\ p^i_{x,t+1} &= p^i_{x,t} + v^i_{x,t} + w^i_{px,t} && (17)\\ p^i_{y,t+1} &= p^i_{x,t} + v^i_{y,t} + w^i_{px,t} && (18)\\ v^i_{x,t+1} &= v^i_{x,t} + u^i_{x,t} + w^i_{vx,t} && (19)\\ v^i_{y,t+1} &= v^i_{y,t} + u^i_{y,t} + w^i_{vy,t} && (20) \end{aligned}$$

In (17), p^{proj} and v^{proj} represent the particle position and the velocity projected onto the discrete cell space, respectively. One could use the same discrete noise for the position and the velocity transitions as defined in (1) - (4), or a continuous noise model with a similar variance.

All the terms in (14) are taken care of by this particle motion, except the cell occupancy state transition distribution $P(S \mid S^{-1}, V^{-1})$. For this term, we have to consider transitions from dynamic occupancies to static occupancies, and vice versa. When a particle transitions from dynamic to static, we set its weight w^{-1}_i to 0 and add the value of w^{-1}_i to the static probability $P(S = s, V = 0 \mid C)$ associated with cell C that the particle moved to. When a static cell transitions to dynamic, on the other hand, new particles are generated in the resampling step.

Finally, if the position of a particle reaches beyond the configuration space $\mathcal{W}$ covered by cells, we stop tracking the particle by setting its weight to 0 since it is out of coverage; it will be eliminated in the resampling step.

[1] The pedestrian motion model defined in Section 2 has an integer velocity, so a real-valued velocity does not necessarily improve the estimation accuracy. However, we could also learn the model from existing data, in which case a real-valued velocity model will be more suitable. For the filtering framework to be as generalizable as possible, it is desired that the velocity is represented in $\mathbb{R}^2$.

3.3.2 Update step for particles

The weight update for each particle is done by multiplying the previous weight w_i^{-1} with the measurement likelihood. Note however that the normalization of the particle weights is done at the same time as the cell static weight normalization (15), since the sum of the particle weights in cell C and the cell static weights needs to equal to one.

$$
\begin{aligned}
w_i =& \frac{P(Z \mid S = d^i, C) w_i^{-1}}{d(C)} && (21) \\
d(C) =& \sum_{SV} P(Z \mid S, C) P(S, V \mid C) \\
=& \sum_{i:(p_x^i, p_y^i) \in C} P(Z \mid S = d^i, C) w_i^{-1} \\
&+ P(Z \mid S = s, C) P(S = s, V = 0 \mid C) \\
&+ P(Z \mid S = e, C) P(S = e, V = 0 \mid C) && (22)
\end{aligned}
$$

3.3.3 Resampling

The resampling step adjusts the number of particles per cell according to the probability that the cell occupancy is dynamic: $1 - P(S = s \vee e \mid C)$. In each cell, the particles with larger weights are more likely to be resampled. In addition, some particles may be newly generated when a static cell transitions to dynamic as discussed in the predict step. In this case, the initial velocities are drawn from a uniform distribution and the initial position is set to the center of the cell.

3.3.4 Global algorithm

The resulting global algorithm is presented in Algorithm 1.

4 Simulation Results

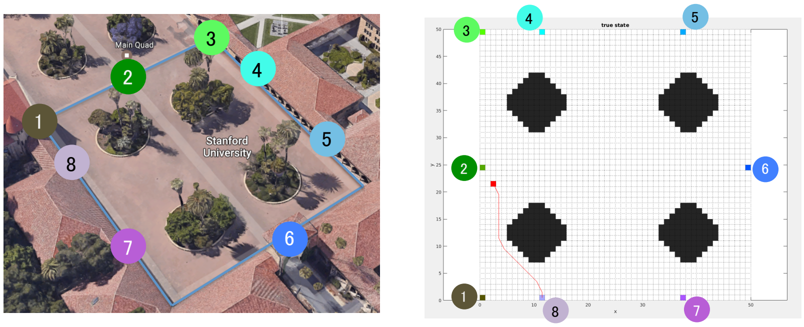
Shown below in Figure 2a is a half of the Stanford University Main Quad, which is simulated with the 50x50 regular grid cells illustrated in Figure 2b.

The sensor measurements are subject to noise as can be seen in Figure 3a, with a false alarm rate and a missed detection rate of 0.1%. The estimated state with three thousand particles is presented in Figure 3b. Although there was one false alarm, The filter correctly tracked both the position and the intention of the pedestrian.

We also performed a quantitative analysis to evaluate the performance of the proposed approach. First, a single pedestrian tracking scenario was considered as a baseline study. The result is shown in the first column of Table 2. We see that both the false alarm rate and the missed detection rate were significantly lower than the sensor error rates. Furthermore, with three thousand particles

```
foreach particle i do
    Apply intention-dependent motion model. (17) - (20)
    if i transitions to static then
        Add particle weight to $P(S = s, V = 0 \mid C)$.
        Set particle weight to 0.
    end
    if particle position is out of $\mathcal{W}$ then
        Set particle weight to 0.
    end
end
foreach cell C do
    Compute $P(S = s, V = 0 \mid C)$ and $P(S = e, V = 0 \mid C)$. (14)(15)
    Compute normalization constant $d(C)$. (22)
    Compute $P(S = s, V = 0 \mid Z, C)$ and $P(S = e, V = 0 \mid Z, C)$. (15)(22)
    foreach particle in cell C do
        Update particle weight. (21)
    end
end
Resample particles over the entire cells as discussed in Section 3.3.3.
```

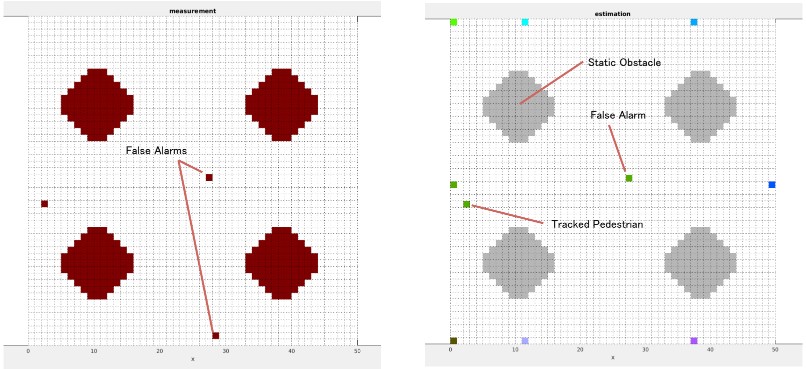
Algorithm 1: Dynamic Occupancy Grid Filtering Algorithm



(a) Stanford Main Quad from Google Maps with 8 possible goal locations marked. The pedestrians are assumed to enter and exit from these goals.

(b) 50x50 grid cell environment. The black cells represent static obstacles. The red cell is a simulated pedestrian with its trajectory shown.

Figure 2: Simulated environment.



(a) Cell occupancy measurements. The brown cells are observed occupied and the white cells empty. The true state was as depicted in Figure 2b. There are two false alarms in this figure due to simulated noise.

(b) Estimated state from the filtering results. The false alarm disappeared at the next time step. The color of the tracked pedestrian indicates the most likely goal location as estimated by the filter.

Figure 3: Measurements and estimation results.

all the pedestrians tracked were labeled with the correct intentions; here the intention inference for a pedestrian is said to be correct if the filter successfully estimates the correct goal location by the time the pedestrian reaches the goal and disappears from the scene.

Max. pedestrians in the scene	1	4	4	4
Particles	3000	3000	6000	12000
Total measurements made	640000	640000	640000	640000
False alarm rate (x10^{-5})	1.41	4.06	4.06	4.21
Missed detection rate (x10^{-5})	2.97	10.4	10.9	8.75
Correct intention inference rate	1.0	0.576	0.758	0.849
Average processing time (s)	0.732	0.712	1.536	3.667

Table 2: Quantitavie performance evaluation for different numbers of maximum pedestrians in the scene at the same time and different numbers of particles.

When the maximum number of pedestrians in the scene at the same time was increased to four, both the false alarm rate and the missed detection rate increased. Nevertheless, they were still orders of magnitude lower than the sensor error rates. It is also worthwhile to note that the correct intention inference rate scaled roughly linearly as we increased the number of particles. Clearly the same three thousand particles was not enough to track four pedestrians simultaneously, but the performance recovered as we added more particles.

5 Conclusions and Future Work

In this paper we have presented a framework for tracking multiple moving pedestrians and estimating their goal locations simultaneously. We have employed an optimal decision making approach to simulate pedestrian motion, and computed a set of optimal policies to provide the filtering framework with the intention-dependent motion model. The estimation algorithm is based on the dynamic occupancy grid filtering method, which does not require any explicit data association to track different intentions of multiple pedestrians. Through the simulation we have demonstrated our algorithm and confirmed that the number of particles required to correctly estimate the goal locations scales linearly with the number of pedestrians in a scene at the same time.

Our next step is to account for object occlusions that are not discussed in this paper, introduce an ego robot with this sensory system into the environment and have it interact with the pedestrians. We are also interested in improving the computational time by parallelizing this algorithm using existing high performance computing techniques.

References

[1] M. Saval-Calvo, L. Medina-Valdés, J. M. Castillo-Secilla, S. Cuenca-Asensi, A. Martínezlvarez, and J. Villagrá, "A Review of the Bayesian Occupancy Filter." *Sensors (Basel, Switzerland)*, vol. 17, no. 2, 2017.

[2] C. Coué, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessiere, "Bayesian Occupancy Filtering for Multitarget Tracking: an Automotive Application," *International Journal of Robotics Research*, vol. 25, no. 1, pp. 19–30, 2006.

[3] H. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI Magazine*, vol. 9, no. 2, pp. 61–74, 1988.

[4] C. Coué, T. Fraichard, P. Bessiere, and E. Mazer, "Using Bayesian Programming for multi-sensor multi-target tracking in automotive applications," in *2003 IEEE International Conference on Robotics and Automation*, 2003, pp. 2104–2109.

[5] A. Negre, L. Rummelhard, and C. Laugier, "Hybrid sampling Bayesian Occupancy Filter," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, 2014, pp. 1307–1312.

[6] L. Rummelhard, A. Negre, and C. Laugier, "Conditional Monte Carlo Dense Occupancy Tracker," in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 2015.

[7] C. Laugier, I. E. Paromtchik, M. Perrollaz, M. Y. Yong, J. Yoder, C. Tay, K. Mekhnacha, and A. Negre, "Probabilistic Analysis of Dynamic Scenes

and Collision Risks Assessment to Improve Driving Safety," *IEEE Intelligent Transportation Systems Magazine*, vol. 3, no. 4, pp. 4–19, 2011.

[8] T. Gindele, S. Brechtel, J. Schroder, and R. Dillmann, "Bayesian Occupancy grid Filter for dynamic environments using prior map knowledge," in *2009 IEEE Intelligent Vehicles Symposium*, 2009, pp. 669–676.

[9] S. Brechtel, T. Gindele, and R. Dillmann, "Recursive importance sampling for efficient grid-based occupancy filtering in dynamic environments," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 3932–3938.

[10] G. Best and R. Fitch, "Bayesian intention inference for trajectory prediction with an unknown goal destination," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 5817–5823.

[11] H. Bai, S. Cai, N. Ye, D. Hsu, and W. S. Lee, "Intention-aware online POMDP planning for autonomous driving in a crowd," in *2015 IEEE International Conference on Robotics and Automation*, 2015, pp. 454–460.

[12] M. J. Kochenderfer, C. Amato, G. Chowdhary, J. P. How, H. J. D. Reynolds, J. R. Thornton, P. A. Torres-Carrasquillo, N. K. Üre, and J. Vian, *Decision Making Under Uncertainty: Theory and Application*, 1st ed. The MIT Press, 2015.