People Finding under Visibility Constraints using Graph-Based Motion Prediction

AbdElMoniem Bayoumi, Philipp Karkowski, and Maren Bennewitz*

Humanoid Robots Lab, University of Bonn, Germany
{abayoumi,philkark,maren}@cs.uni-bonn.de

Abstract. An autonomous service robot often first has to search for a user to carry out a desired task. This is a challenging problem, especially when this person moves around since the robot's field of view is constrained and the environment structure typically poses further visibility constraints that influence the perception of the user. In this paper, we propose a novel method that computes the likelihood of the user's observability at each possible location in the environment based on Monte Carlo simulations. As the robot needs time to reach the possible search locations, we take this time as well as the visibility constraints into account when computing effective search locations. In this way, the robot can choose the next search location that has the maximum expected observability of the user. Our experiments in various simulated environments demonstrate that our approach leads to a significantly shorter search time compared to a greedy approach with background information. Using our proposed technique the robot can find the user with a search time reduction of 20% compared to the informed greedy method.

1 Introduction

Finding a person is an essential functionality that is needed by several applications of mobile service robots. Typically, users do not stay at a fixed position but move along common paths between places where they remain for a while, e.g., to discuss work with a colleague, grab some material, or get a coffee. The robot needs a good strategy to find the user as fast as possible also in these situations to carry out its task.

One possible solution to the search problem is to apply techniques that try to maximally cover the visible area of the environment [1–3]. However, these approaches often lead to long search times and high navigation costs as they aim

* All authors are with the Humanoid Robots Lab, University of Bonn, Germany. This work has been supported by the German Academic Exchange Service (DAAD) and the Egyptian Ministry for Higher Education as well as by the European Commission under contract number FP7-610532-SQUIRREL and by the DFG Research Unit FOR 2535 Anticipating Human Behavior.

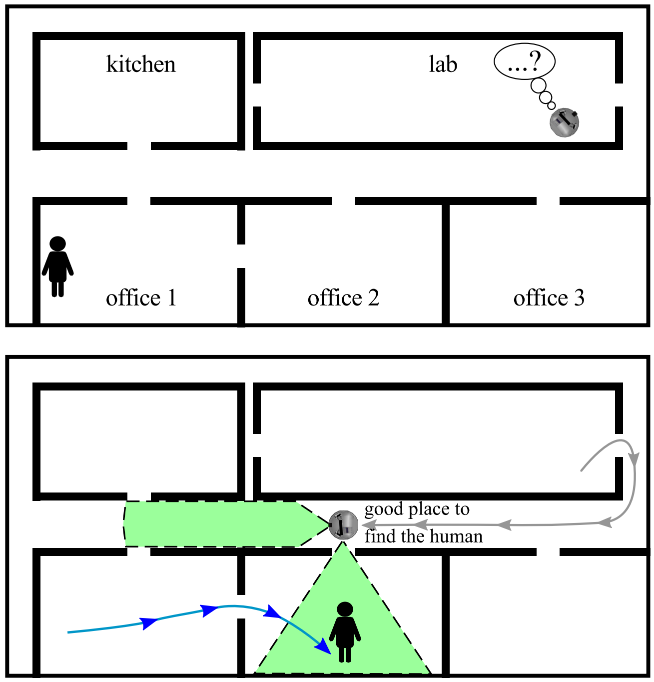

Fig. 1. Top: The robot needs to find a user whose current location is unknown. The user may walk toward any of a set of predefined destinations, known by the robot. Bottom: The robot needs to select a good search location that covers most of the expected paths of the user. Our approach selects a search location with maximum observability of the user at the time the robot reaches it.

at covering the whole environment. Moreover, the maximum coverage techniques will not necessarily revisit already covered regions, which might be necessary since the user is assumed to move across the environment and the robot might miss him during the search.

In this paper, we make use of prior knowledge about frequently visited destinations of the user and their connecting paths. We developed an approach that determines good search locations using a particle filter based prediction model on a graph representation of paths the user typically takes. Our novel approach computes the likelihood of the observability of the user at each possible location based on Monte Carlo simulations. We hereby take into account the time needed by the robot to reach the search locations from its current position as well as the visibility constrains that arise from the robot's limited field of view and obstacles. Fig. 1 highlights the strength of our approach. The location of the user is initially unknown. Our approach leads to the selection of an effective search location that provides the highest expected observability, i.e., the robot can observe the corridor and the entrances to multiple rooms and, thus, locate the user.

We show in extensive simulated experiments and in various environments that our technique generates search locations that significantly reduce the time to find the user compared to a greedy solution that is provided with background information about the possible destinations between which the user moves. In the experiments, we model noisy observations and dynamic obstacles to show the robustness of our approach.

2 Related Work

The problem of finding a moving person in an environment was early studied as a coverage problem based on the robot's visibility polygon [1–5]. Stiffler *et al.* [6] additionally considered the problem of unreliable sensors in this context. The authors developed a visibility-based geometric formulation to place the surveillance robot at specific environment locations that maximize the path of the intruder through the robot's visible region to increases the likelihood of observing the intruder. All these solutions to the coverage problem do not predict motion of the person and thus cannot provide any pose estimate. They lead to long search times and high navigation costs as they aim at covering the whole environment.

On the other hand, several approaches that predict motions and aim at minimizing the searching time for a mobile robot have been presented. Tipaldi and Arras [7] proposed to learn a spatial affordance map and apply a Poisson process to relate space, time, and occurrence probability of activity events. Afterwards, the spatio-temporal model can be used to generate an optimal path on a grid map of the environment for a mobile robot to encounter specific humans. This approach does not make use of any sensor modalities to update the believe about the location of a user but considers just the encounter probability of grid cells. Schwenk *et al.* [8] developed a search approach that uses a highly abstract topological representation of the environment and learns about the user's behaviors in order to estimate the likelihood of the user's current room. Here, it is assumed that the robot detects people when they are within a range of 1.8 m around the robot. Kulich *et al.* [9] introduced a model that learns the temporal likelihood of possible desired interactions to actively search for humans in order to interact with them in public space. Krajník *et al.* [10] presented a method based on spatio-temporal models to enable the robot finding non-stationary objects in an office environment. The authors represent the environment as an abstract topological map and combine it with periodic functions in order to compute the likelihood of existence of the objects at any node of the map with respect to the time. All these approaches, however, ignore the visibility constraints resulting from the environment layout.

Other approaches considered the frequency of human existence at specific locations. The idea here is to construct a probability distribution for every hour of the day. For example, in the work of Volkhardt and Gross [11], the robot searches for the human at pre-defined locations, where each location is assigned a probability relative to the frequency of observing the human there. Accordingly, the robot selects the location with the highest probability. Mehdi and Berns [12] presented a technique that generates a minimum set of view points that ensure a maximum coverage of the environment with the robot's constrained field of view. The authors proposed to construct a probability distribution about the human's observability at these destinations during each hour of the day and take the navigation cost into account for deciding which one of the view points to chose as search location. These methods do not model the human's motion and therefore cannot predict their expected position at a certain intermediate time step.

Goldhoorn *et al.* [13] proposed using particle filters to estimate the most likely location of the user at the current time step. The robot moves toward that location for few time steps then updates its estimate about the user's position and recomputes the robot's movement. As opposed to our method, this technique does not take into account the time needed by the robot to reach search locations from its current place. Moreover, moving the robot just for few time steps and then selecting another search location often lead to oscillating navigation behavior as the estimation jumps across the map as we realized in our experiments.

In contrast to all the mentioned search methods, our system models the human's motion and provides a probability distribution about his/her position at each time step. We consider the robot's limited field of view and visibility constraints when computing the likelihood of observing the user at a certain place and also take into account the time needed by the robot to reach the search locations.

3 Problem Formulation

The task of the robot is to find a non-stationary user as fast as possible. The environment is hereby known to the robot and it has prior knowledge about locations where the user frequently stays and his/her typical paths between these locations. We refer to these locations as *destinations*. After reaching such a destination, the user might stay there or move to another destination after a certain waiting time.

We represent the environment as a grid map with an overlaid topo-metric graph as shown in Fig. 3, where each cell in that grid is mapped onto its closest graph node [14]. The connections shown between neighboring nodes correspond to valid paths between these nodes. However, some of the paths are only passable by humans, e.g., due to the size of the robot or any other potential constraints of the searching environment.

The location of the user is initially unknown to the robot as well as his/her intended destinations when moving. After reaching a destination, the user might stay there or start moving to another destination after some time. We assume the moving velocity of the user to be within a certain range, however, the exact velocity of the user is unknown to the robot. Dynamic obstacles, e.g., other humans, can appear in the environment and temporarily constrain the robot's field of view. The task is considered as successful when the robot observes the user within its field of view.

4 Graph-Based People Tracking

To represent the belief about the location of the user and track its motion on the graph between the destinations, we apply a particle filter, inspired by the work of Liao *et al.* [15].

We use the information about the typical paths between the destinations and the times the user stays at the destinations to find the average time that the user occupies each node. We sample the pose of the particles according to this occupation likelihood. For each particle, we independently sample one of the destinations as the next "goal" based on the typical paths that lead through its node.

Each particle then moves to a graph node along the path to its destination according to a Gaussian motion model, taking into account the velocity range of the user. Whenever a particle reaches its destination (or is initialized at a destination), it remains there for a sampled time interval that corresponds to the typical waiting behavior of the user. Finally, we select another destination for the particle as its next goal according to the transition probability:

$$p(Dest_i = D_b | n_i = D_a) = p(D_b | D_a), \tag{1}$$

where $Dest_i$ is the chosen destination of particle i, n_i is the graph node of particle i, D_a and D_b are two destinations, and $p(D_b|D_a)$ is the known probability of the user to move from D_a to D_b.

The particles are weighted proportional to the observation likelihood. The weights are initially set to the same value and then updated at each time step as follows. For particles that fall within the robot's field of view while the user is not currently detected, the weights are reduced:

$$w_i = \begin{cases} \gamma w_i, & \text{if } (n_i \in \mathcal{FOV}) \land \textit{user not detected} \\ w_i & \text{otherwise} \end{cases}, \tag{2}$$

where w_i is the weight of particle i, n_i is again the current graph node of particle i, $\mathcal{FOV}$ is the area covered by the robot's visual sensors and $\gamma \in [0, 1)$ is a reduction factor. Since the likelihood of false negative observations increases with the distance of the user to the robot, γ decreases with this distance.

As we assume a proper identification system, we do not model false positive observations. Note, however, that we can deal with false positive observations for a short time by requiring a minimum number of subsequent time steps where the human is detected before the search is assumed to be successful.

5 Selecting Search Locations via Monte Carlo Simulations

In this section, we describe our approach to selecting effective search locations for the robot to find the human. Relying only on the estimated most likely location of the user at each time step leads to an oscillating navigation behavior as the estimation might jump across the map. We, therefore, propose a method based on Monte Carlo simulations that takes into account the time needed by the robot to reach the possible search locations from its current place.

We first perform Monte Carlo simulations to compute the positions of the particles at future time steps according to the motion model. In particular, we simulate the particle propagation along the graph according to the motion model as many future time steps as needed by the robot to reach the furthest graph node relative to the robot's current node. We then compute the likelihood of the user's observability at each graph node, while considering the time needed to reach this node. For example, if a node lies ten time steps away, we consider the simulated particle distribution ten time steps into the future when computing likelihood of the user's observability at this node. The weights of the simulated particles stay unaffected during the Monte Carlo simulations.

We compute the observability likelihood l_j of the user at each node j as follows

$$l_j = \sum_{i \in \mathcal{O}_j^t} w_i, \quad \forall j \in \mathcal{R}^t, \quad 1 \leq t \leq T, \tag{3}$$

where $\mathcal{R}^t$ is the set of graph nodes that can be reached from the robot's current node n_r within exactly t future time steps, T is the number of future time steps needed to reach the furthest graph node from n_r, and $\mathcal{O}_j^t$ is the group of simulated particles at future time step t that can be observed from node j.

After computing l_j for every j, we select the graph node with the highest observability likelihood s as the next search location[1], i.e.,

$$s = \operatorname*{argmax}_j l_j. \tag{4}$$

The pseudo-code of our search goal selection algorithm is listed in Alg. 1. As can be inferred from the example shown in Fig. 2, the robot selects the next search goal as the location that provides highest observability at the time the robot reaches it.

The robot then navigates to the selected node along the shortest path in the graph and does an observation action by performing a full rotation. If the user cannot be found anywhere on the way to the current search location nor while performing the observation action, a new search location is selected as previously and so forth. Performing the particle simulations in the described way and including them in the calculation of the observation likelihood provides an effective method for selecting a good search location that takes into account the dynamic behavior of the user.

While computing the next search location, we do not consider waiting actions or non-shortest paths, as this results in infinite possibilities to reach any node. Neither do we take into account the observability along the intermediate nodes

[1] Note that the time is inherently considered in the computation of the l_j, such that s does not need to have a time index.

Algorithm 1: Selection of the next search location using Monte Carlo simulations

Input : $particles$ and $robotPose$
Output: next search location
$likelihood \leftarrow \{\}$;
for $t \leftarrow 1$ **to** T **do**
  $particles \leftarrow$ simulate $particles$ one step ahead acc. to the motion model;
  $reachableNodes \leftarrow$ nodes that can be reached by the robot in exactly t time steps;
  $nodesWeights \leftarrow \{\}$;
  // calculate collective weight for each node at time step t;
  for $i \leftarrow 1$ **to** $particles.size$ **do**
    $node \leftarrow particles[i].node$;
    $weight \leftarrow particles[i].weight$;
    $nodesWeights[node] \leftarrow nodesWeights[node] + weight$;
  end
  // calculate observability likelihood for each node;
  for $r \in reachableNodes$ **do**
    $visibleNodes \leftarrow$ visible nodes from r (incl. r) at time step t;
    for $v \in visibleNodes$ **do**
      $likelihood[r] \leftarrow likelihood[r] + nodesWeights[v]$;
    end
  end
end
return $\operatorname{argmax}_{node}\ likelihood[node]$;

to the considered search location. As we have found out in our experiments, this leads to a selection of search locations with longer paths and does not decrease the search time.

6 Experimental Results

We carried out extensive experiments to evaluate our approach and compare it to alternative methods.

6.1 Experimental Setup

We performed the experiments in three different, challenging simulation environments (see Fig. 3), each of size 41 m $\times$ 20.5 m with a grid map resolution of 0.25 m and a node distance of 1.5 m. In the first two environments, multiple paths exist between the destinations, among which the user chooses one based on a certain known probability distribution and the transition probabilities from one destination to the others are equally likely (see Eq. 1). Note, however, that some

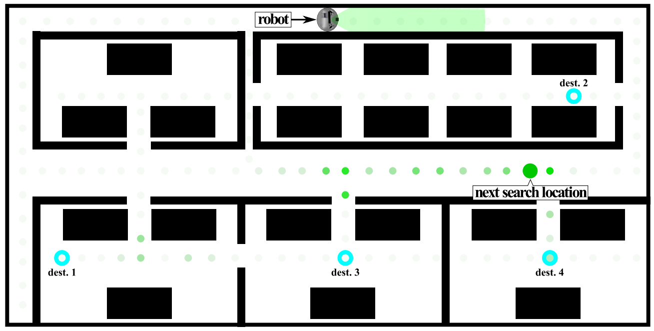

Fig. 2. This figure shows the selected search location according to Alg 1. The graph nodes are drawn with a color intensity corresponding to the observability likelihood of the user at the time the robot reaches this search location. The robot selects the node that provides the highest observability likelihood as next search location.

passages are impassible to the robot, i.e., the dotted line with orange nodes for the first two environments to make the search problem even more challenging.

In each experiment, the position of the user is initialized according to the occupation likelihood (see Sec. 4) and the user moves in the environment between the predefined destinations. The user does not necessarily move on the shortest path but might take detours. When the user reaches his/her destination, he/she waits there for a certain period of time. The user repeats this behavior until he/she reaches his/her fourth destination and remains there. The velocity of the user is sampled from a certain interval. At each time step, the position of the user is mapped onto the closest graph node given its grid map position. The initial location of the user is unknown to the robot and is outside its field of view.

We use 150 particles to represent the belief about the position of the user and track it on the graph representation of the environment. The particles are initialized and updated as described in Sec. 4.

The number of dynamic obstacles that constrain the robot's field of view ranges from three to five and their velocities are sampled from the same interval as the velocity of the user.

The search task is considered successful when the robot observes the user. The robot's field of view has a horizontal opening angle of 58°, which corresponds to that of an *ASUS Xtion Pro Live*, and a 10 m view distance. We set the probability of false negatives between 0.05 and 0.15 linearly increasing with the distance between the robot and the user. We do not consider false positive observations in the simulation experiments.

6.2 Evaluation and Results

We performed $5,000$ experiments in each of the three environments. In order to evaluate the performance of our approach, we considered the search time and

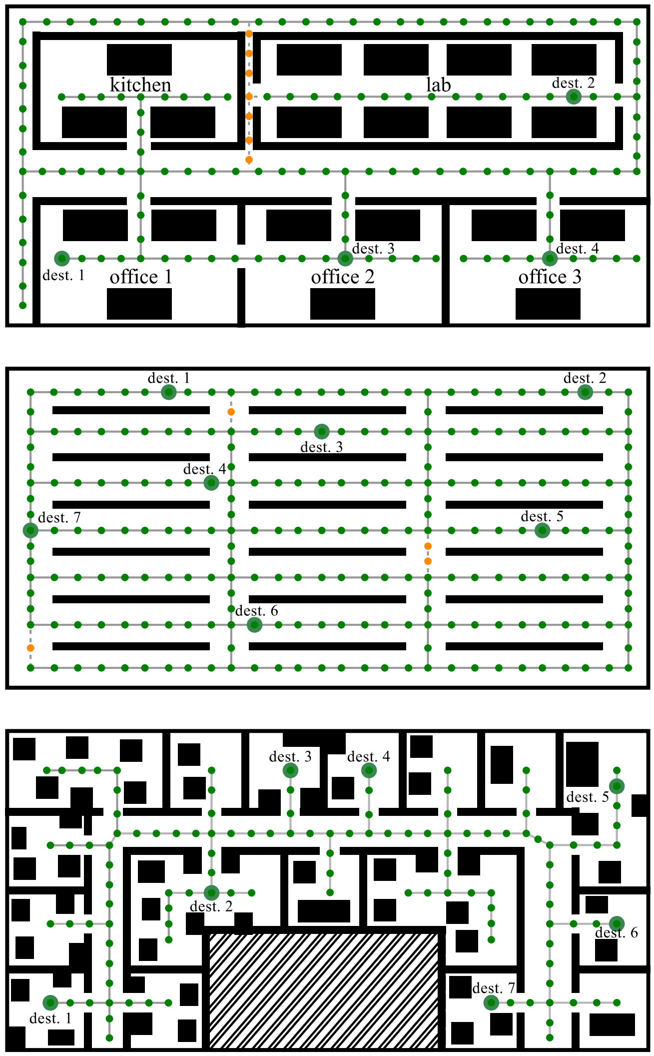

Fig. 3. Three simulation environments with overlaid topo-metric graphs. Each environment is represented as a grid map with an overlaid graph, where each grid cell is mapped to the closest graph node (green dots) in the same room. The orange dots represent paths that are only passable by the user but not by the robot. The bold green dots represent the predefined destinations between which the user moves.

compared it to the time needed by two different approaches. The strategy of the first alternative approach is to visit all destinations in a greedy fashion using background information, i.e., the knowledge about the destinations of the user. The greedy approach does not consider any prediction about the user's location; it keeps selecting the closest unvisited destination as a search location until the

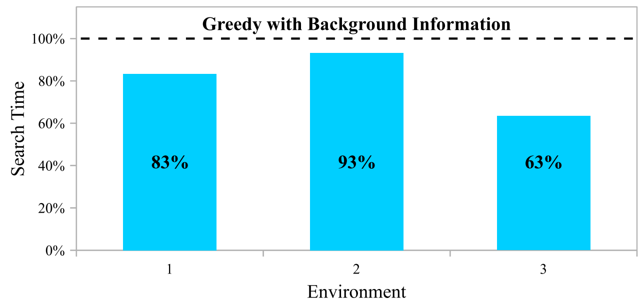

Fig. 4. Average relative search time achieved by our approach with respect to the greedy approach with background information. The times are normalized so that the greedy approach equals 100%.

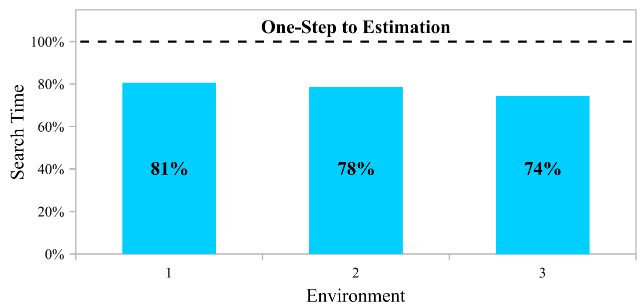

Fig. 5. Average relative search time achieved by our approach with respect to the one-step to estimation method. The times are normalized so that the one-step to estimation approach equals 100%.

Table 1. Percentage of switching to the greedy approach.

	Our Approach	One-Step to Estimation
Env. 1	2%	7.38%
Env. 2	1.14%	5%
Env. 3	7.45%	15.05%

user is found. After visiting all destinations it starts the search process all over again.

We additionally compared our approach to a method that uses the particle filter representation to infer the currently most likely location of the user and moves the robot toward that location for one time step, then updates the estimation and so on. This method is similar to the approach of Goldhoorn *et al.* [13] and we refer to this method as the *one-step to estimation* method.

We evaluated the statistical significance of our comparative experiments with a *two-tailed paired t-test.* The experimental results show that our method performing Monte Carlo simulations significantly outperforms each of the other two approaches with a statistical significance of 99%. Fig. 4 and Fig. 5 show the average relative search times achieved by our approach for each of the three environments with respect to the greedy approach with background information and the one-step to estimation method, respectively.

As our approach does not guarantee to cover the entire map and, thus, might not find a search location close to the user's final destination, we switch to the greedy approach after a given time limit. The maximum time limit was determined experimentally such as to minimize the overall search time. Table 1 shows the percentage of experimental runs using our approach and the one-step to estimation" approach that exceeded this time limit and switched to the greedy method. As shown, our technique based on Monte Carlo simulations outperforms the one-step to estimation method for all the environments.

A video showing the advantages of our approach for an example run can be downloaded from https://www.hrl.uni-bonn.de/ias18bayoumi.mp4.

7 Conclusion

In this paper, we presented an approach that enables a mobile robot to quickly find a non-stationary user in complex environments. Our method selects the next search location by predicting future paths of the user. To compute the likelihood of the observability of the user at possible search locations, we apply Monte Carlo simulations using a particle filter on a graph representation of possible paths in the environment. We hereby take into account the time needed by the robot to reach the search locations as well as visibility constraints.

As our simulation experiments demonstrate, our approach enables the robot to select effective search locations to find the user within a short amount of time. We showed in extensive experiments that our proposed method significantly outperforms two other common search methods. The experiments included runs where occlusions caused by dynamic obstacles as well as false negative detection occurred, which will be the case for real-world scenarios.

References

1. Suzuki, I., Yamashita, M.: Searching for a mobile intruder in a polygonal region. SIAM Journal on Computing **21**(5) (1992) 863–888
2. Guibas, L., Latombe, J., LaValle, S., Lin, D., Motwani, R.: A visibility-based pursuit-evasion problem. International Journal of Computational Geometry and Applications (1996) 471–494
3. Choset, H.: Coverage for robotics – a survey of recent results. Annals of Mathematics and Artificial Intelligence **31**(1) (2001) 113–126
4. Moors, M., Rohling, T., Schulz, D.: A probabilistic approach to coordinated multi-robot indoor surveillance. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS). (2005) 3447–3452

5. Kolling, A., Carpin, S.: Multi-robot surveillance: An improved algorithm for the graph-clear problem. In: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA). (2008) 2360–2365
6. Stiffler, N., Kolling, A., O'Kane, J.: Persistent pursuit-evasion: the case of preoccupied pursuer. In: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA). (2017) 5027–5034
7. Tipaldi, G., Arras, K.: I want my coffee hot! Learning to find people under spatio-temporal constraints. In: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA). (2011) 1217–1222
8. Schwenk, M., Vaquero, T., Nejat, G., Arras, K.: Schedule-based robotic search for multiple residents in a retirement home environment. In: Proc. of the National Conf. on Artificial Intelligence (AAAI). (2014) 2571–2577
9. Kulich, M., Krajník, T., Přeučil, L., Duckett, T.: To explore or to exploit? Learning humans' behaviour to maximize interactions with them. In Hodicky, J., ed.: Modelling and Simulation for Autonomous Systems: Third International Workshop, MESAS 2016, Revised Selected Papers. Springer International Publishing (2016) 48–63
10. Krajník, T., Kulich, M., Mudrová, L., Ambrus, R., Duckett, T.: Where's Waldo at time t? Using spatio-temporal models for mobile robot search. In: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA). (2015) 2140–2146
11. Volkhardt, M., Gross, H.: Finding people in home environments with a mobile robot. In: Proc. of the European Conf. on Mobile Robots (ECMR). (2013) 282–287
12. Mehdi, S., Berns, K.: Behavior-based search of human by an autonomous indoor mobile robot in simulation. Universal Access in the Information Society **13**(1) (2014) 45–58
13. Goldhoorn, A., Garrell, A., Alquézar, R., Sanfeliu, A.: Searching and tracking people in urban environments with static and dynamic obstacles. Robotics and Autonomous Systems **98** (2017) 147–157
14. Bayoumi, A., Karkowski, P., Bennewitz, M.: Learning foresighted people following under occlusions. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS). (2017)
15. Liao, L., Fox, D., Hightower, J., Kautz, H., Schulz, D.: Voronoi tracking: Location estimation using sparse and noisy sensor data. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS). Volume 1. (2003) 723–728 vol.1