Speeding Up Person Finding Using Hidden Markov Models

AbdElMoniem Bayoumi[a,b,*], Philipp Karkowski[a], Maren Bennewitz[a,**]

[a] *Humanoid Robots Lab, University of Bonn, Computer Science VI, Endenicher Allee 19A, 53115 Bonn, Germany*
[b] *Department of Computer Engineering, Faculty of Engineering, Cairo University, Gamaa Street, 12613 Giza, Egypt*

Abstract

The ability of mobile service robots to efficiently search for a person is needed in a vast domain of applications. The search problem is especially challenging when the user is freely moving across the environment, the robot has only a constrained field of view, and visibility constraints arise from the environment. We propose in this article a novel approach that simulates the user's presence at different locations in the environment based on a hidden Markov model (HMM). The HMM predicts the user's motion and computes the observability likelihood at the different locations given the predictions. Our approach then selects effective search locations that maximize the user's expected observability. The selection criterion hereby considers the visibility constraints along the robot's path as well as the robot's travel time to reach the search location. We performed both real-world and extensive simulated experiments to evaluate our method. In comparison to a greedy maximum coverage approach as well as to a greedy strategy that uses background information, we show that our framework leads to a significant reduction of the time needed to find the user.

Keywords: Modeling human motion, motion prediction, hidden Markov models, information gain based search

[*] Corresponding author
[**] Principal corresponding author
Email addresses: `abayoumi@cs.uni-bonn.de` (AbdElMoniem Bayoumi), `philkark@cs.uni-bonn.de` (Philipp Karkowski), `maren@cs.uni-bonn.de` (Maren Bennewitz)

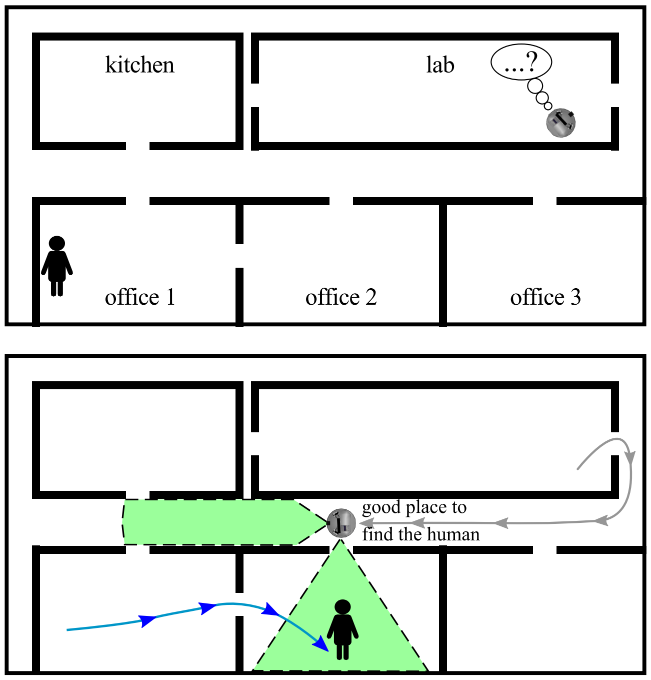

Figure 1: Top: The robot needs to interact with the user, however, the user's current location is unknown. Furthermore, the user may not stay at their current location, instead, they may move between a set of predefined destinations. Bottom: The robot needs to select an effective search location that minimizes the search time of finding the user. Our approach selects a good search location that maximizes the likelihood of observing the user via covering most of the user's expected paths at the time the user is expected to reach them.

1. Introduction

Finding a person is essential for mobile service robots in several applications, e.g., office delivery, household assistance, ... etc. In such scenarios, the robot often has to assist the user with tasks that involve direct interactions as well as free moves across the environment. Typically, users do not permanently stay at fixed locations, instead, they frequently move along common paths between designated locations, e.g., to grab a coffee, use the printer, or talk to colleagues. Therefore, a good search strategy is needed to enable the robot to quickly find the user in such situations if necessary.

In this article, we introduce a novel approach for selecting search locations by performing simulations based on a hidden Markov model (HMM) to predict possible behaviors of the user. Our method simulates the user's motion and computes the predicted belief about the user's presence at the possible locations for future time steps. We then use the predicted belief, while also considering the visibility characteristics of the environment to iteratively select good search locations with a high likelihood of observing the user. Thereby, our approach takes into account the time needed by the robot to reach a certain search location, which is essential for the robustness of the prediction. Figure 1 highlights the strength of our approach. As illustrated, our proposed approach selects an efficient search location with a high probability of observing the user.

As we show in extensive simulated experiments in various environments, our approach enables a robot to select good search locations, which leads to a significantly reduced search time compared to a greedy maximum coverage approach and another greedy method that uses background information. The greedy maximum coverage approach chooses search locations that maximize coverage of so far unobserved areas, however, the other greedy method uses background information about the destinations at which the user frequently stays. For a realistic setting, we model noisy observations and dynamic obstacles and show the robustness of our approach.

In contrast to our previous work [1], we apply a HMM for the motion prediction instead of a particle filter to achieve a more robust performance, because the particle filter depends on the random initialization and propagation of particles. We found throughout our experiments that re-running the experiments of the particle filter approach multiple times sometimes leads to varying results due to the randomness. This variance in the performance decreases as the number of particles increase. Accordingly, using an infinite number of particles will rule out this variance which is equivalent to using HMM. Additionally, we compute the user's observability likelihood along the robot's path to the potential search locations in order to have a more accurate estimate of the probability to encounter the user for every search location. Moreover, we determine key locations along the robot's path to the selected search location at which performing observation actions will increase its chances of encountering the user, e.g., hall way intersections, office entrances, etc. Furthermore, we carried out a real-world experiment to demonstrate the practical applicability of our search strategy.

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. One possible solution to solve this problem is to apply techniques that aim at maximally covering the visible area of the environment, e.g., Suzuki and Yamashita [2], as well as Guibas *et al.* [3] tackle the problem of finding users by a surveillance robot. They make use of the geometrical properties of the robot's visibility polygon in order to find a directed graph on which the robot takes a tour to clear the environment from unpredictable dynamic user. The authors use the geometric properties of the visibility polygon in order to minimize the visited points through the environment. However, they do not predict motion of the user and thus do not estimate the user's pose. On the other hand, Lee *et al.* [4] consider a one-room version of the visibility polygon problem, where the robot with a limited field of view is located in only one room and the user may be present in the room or may walk outside. The robot has to cover all the area of room using its limited field of view to find the user. Additionally, Isler *et al.* [5] apply a randomized coverage approach to find a dynamic user by representing the environment as a tree using triangulation and determining which leaf to visit via guessing. The authors show that their approach outperforms deterministic maximum coverage. Furthermore, a survey about different coverage approaches is presented by Choset [6].

Stiffler *et al.* [7] additionally considered the problem of unreliable sensors in this context. The authors developed a visibility-based geometric formulation to place a surveillance robot at specific environment locations that maximize the user's expected path through the robot's visible region to increases the likelihood of observing the user. All these solutions to the coverage problem do not predict the motion of the person and may lead to long search times with high navigation costs as they aim at covering the whole environment.

Several approaches that use multi-robot collaboration have been presented, e.g., Moors *et al.* [8] use a topometric graph representation to assign a team of robots as guards at vital connections in the environment, i.e., doors and corridors, to avoid recontamination of cleared areas. Gerkey *et al.* [9] propose a similar way of coordination between robots of limited field of view to guard each other in order to avoid missing the user. Moreover, Kolling and Carpin [10] solve the graph-clearance problem in NP-complete way using multiple robots. Hollinger *et al.* [11] assume that the user's motion model is given. Therefore, some robots make use of the given information while others perform maximum coverage to balance between robustness and efficiency. All these coverage approaches often lead to long search times and high navigation costs as they aim at covering the complete environment and neglect background information about typical user behaviors.

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 [12] 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 belief about the location of a user but considers just the encounter probability of grid cells. Schwenk *et al.* [13] developed a schedule-based search approach that uses a highly abstract topological representation of the environment and learns about the abstract behaviors of users in order to estimate the likelihood of the users current rooms. Kulich *et al.* [14] introduced a model that learns the temporal likelihood of desired interactions to actively search for humans in order to interact with them in public space. Krajník *et al.* [15] 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 presence 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 evaluate the frequency of user's presence at specific locations. The idea here is to construct a probability distribution for every hour of the day. For example, Volkhardt *et al.* [16] search for users at predefined locations at which they frequently stay. Then, in the work of Volkhardt and Gross [17], each predefined location is assigned a probability relative to the frequency of observing the user there. Accordingly, the robot selects the location with the highest probability. Mehdi and Berns [18] 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 user'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 user's motion and therefore cannot predict their expected position at a certain intermediate time step.

Goldhoorn *et al.* [19] 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

We consider a scenario where the robot is fully aware of the environment structure and the static obstacles. The environment is represented as a grid map with an overlaid topometric graph as shown in Figure 2 and Figure 3, where each grid cell is mapped to the closest graph node in the same room. Both the robot's and the user's locations are mapped to these nodes.

The robot's goal is to minimize the search time needed to find the moving user whose initial location and intended destination are unknown. The user moves along common paths between a set of possible destinations, which are known to the robot. After reaching one of those destinations, the user might stay there for a while or start moving to another destination after some time.

The robot observes the environment with its camera, which has a constrained field of view. During the search, occlusions can also occur due to static or dynamic obstacles.

4. Graph-Based Person Tracking using Hidden Markov Models

We use a hidden Markov model to represent the belief about the user's location on the graph. Each hidden state corresponds to one of the topometric graph nodes that represent the environment (see Figure 2 and Figure 3). In order to generate the initial belief about the user's location, we first compute the user's average occupation time for each graph node based on the typical time the user spends at the destinations and moves between them. Furthermore, we learn the state transition probability distribution based on the user's common paths between the predefined destinations. For each node, we learn the transition probabilities to the neighboring nodes based on the set of paths that lead through that node. Moreover, this transition probability distribution

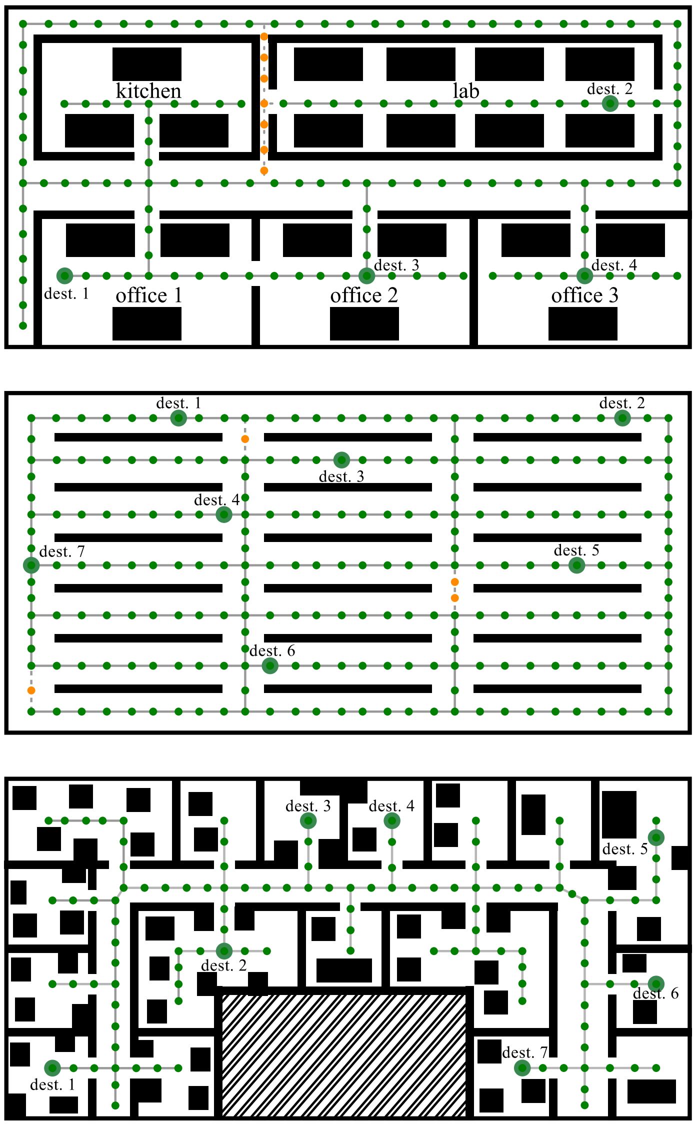

Figure 2: Three simulation environments with overlaid topometric graphs. We use a grid map representation of the environment with an overlaid graph. Each cell in the grid is mapped to the closest graph node (green dots) within the same room. The impassible paths to the robot are represented as orange dots and the bold green dots represent the predefined destinations between which the user moves.

takes into account the average velocity of the user as well as the typical time the user spends at the destinations.

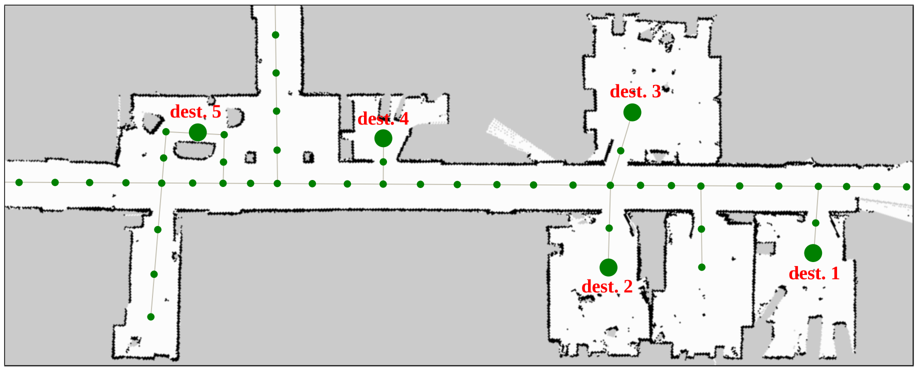

Figure 3: A real indoor environment, representing the Humanoid Robots Lab at the University of Bonn, with an overlaid topometric graph. The bold green dots represent the predefined destinations between which the user moves.

Computing the initial belief of each node and the state transition probability distribution corresponds to the training phase of the HMM. The training process relies on the assumption that a set of sample user paths are given. Depending on the transition probabilities of the destinations, which can be derived from those paths, we compute the initial probabilities of the nodes on the paths as well as their transition probabilities.

At each time step, we update the belief based on the transition probabilities and the belief at the previous time step as follows:

$$Bel_t(i) = \sum_j p(i \mid j)\ Bel_{t-1}(j), \quad \forall i \in \mathcal{N}, \tag{1}$$

where $Bel_t(i)$ is the belief of being at node i at the time step t and $p(i \mid j)$ is the transition probability from node j to node i. Additionally, $\mathcal{N}$ is the set of graph nodes.

After that, we update the belief of the graph nodes proportional to the observation likelihood. According to our problem, positive observation of the user leads to a successful termination of the search process. Therefore, we only consider negative observations in the observation model to update the belief. For the graph nodes that fall within the robot's field of view while the user is not currently detected, the probability is reduced, as follows:

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

where $\mathcal{FOV}$ 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

In this section, we describe our approach to select 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, following the approach of Goldhoorn *et al.* [19], leads to an oscillating navigation behavior as the estimation often jumps across the map as we noticed in our experiments. We, therefore, propose a method that performs HMM-based simulations and takes into account the time needed by the robot to reach the possible search locations to find the node that provides the highest likelihood of observing the user.

5.1. Motion Prediction

We first perform HMM-based simulations to compute the likelihood of the user's presence for each graph node at future time steps. Thus, we use the HMM presented in the previous section in order to compute this likelihood. We predict the human's motion along the graph by simulating the state transition for future time steps and compute the probabilities of the graph nodes according to Equation 1. The belief at any time step represents the likelihood of the user's presence at the corresponding graph node at that time step.

5.2. Observability Likelihood

After that, we use the predicted belief distributions to compute the likelihood of observing the user from each graph node, i.e., the likelihood of the user's observability at that node, while considering the time needed to reach this node. For example, if a node lies ten time steps away, we consider the computed belief resulting from simulating the state transitions ten time steps into the future when computing the likelihood of the user's observability at this node.

One approach is to compute the observability likelihood l_k of the user at each node k as follows:

$$l_k = \sum_{i \in \mathcal{O}_k^{\mathcal{T}}} Bel_{t+\mathcal{T}}(i), \quad \forall k \in \mathcal{R}^{\mathcal{T}}, \quad 1 \leq \mathcal{T} \leq T, \tag{3}$$

where $\mathcal{R}^{\mathcal{T}}$ is the set of graph nodes that can be reached from the robot's current node n_r within exactly $\mathcal{T}$ future time steps, T is the number of future time steps needed by the robot to reach the furthest graph node from n_r, and $\mathcal{O}_k^{\mathcal{T}}$ is the group of graph nodes that can be observed from node k by performing an observation action, i.e., a full rotation. Moreover, $Bel_{t+\mathcal{T}}(i)$ is the predicted belief about the user's presence at node i and time $t+\mathcal{T}$.

5.3. Adding Observability Along the Path

Furthermore, we considered an extension of the computation of the user's observability likelihood in Equation 3 such that we additionally take into account the user's observability likelihood at each of the intermediate nodes along the robot's path, i.e., starting from its current location on the shortest path to the possible search goal, instead of only considering this likelihood at the selected node itself. However, this leads the robot to favor far search locations in order to maximize the observability likelihood accumulated along the path. To overcome this issue, we evaluate the average observability likelihood for the intermediate nodes instead of their summed individual likelihoods.

Additionally, computing the observability likelihoods along the robot's path based on the simulated HMM belief will suffer from some inaccuracy due to observing the same nodes for multiple successive time steps when simulating the robot's journey. In order to alleviate this issue we apply a reduction factor to the computed average observability likelihoods of the intermediate nodes. This reduction factor, which is computed independently for each path, corresponds to the ratio between the number of the observed intermediate nodes without repetition to that with repetition. For example, this ratio takes the value of 1 only if each observed intermediate node is only observed once during the robot path to its search location. Accordingly, we compute the observability likelihood so that it takes into account observability along the robot's path:

$$\begin{aligned} l'_k &= l_k + \frac{\alpha}{\mathcal{T}-1}\sum_{\delta=1}^{\mathcal{T}-1}\sum_{i\in\mathcal{Q}_k^{t+\delta}} Bel_{t+\delta}(i) \\ &= \sum_{i\in\mathcal{O}_k^{\mathcal{T}}} Bel_{t+\mathcal{T}}(i) \ + \frac{\alpha}{\mathcal{T}-1}\sum_{\delta=1}^{\mathcal{T}-1}\sum_{i\in\mathcal{Q}_k^{t+\delta}} Bel_{t+\delta}(i), \\ &\qquad \forall k\in\mathcal{R}^{\mathcal{T}},\ 1\leq\mathcal{T}\leq T, \end{aligned} \tag{4}$$

where

$$\alpha = \frac{\sum_{\delta=1}^{\mathcal{T}-1}\sum_{i\in\mathcal{Q}_k^{t+\delta}}\beta_i}{\sum_{\delta=1}^{\mathcal{T}-1}|\mathcal{Q}_k^{t+\delta}|}, \tag{5}$$

and

$$\beta_i = \begin{cases} 1 & \text{if first time to observe } i \\ 0 & \text{otherwise} \end{cases}. \tag{6}$$

Here, $\mathcal{O}_k^{\mathcal{T}}$ is the group of graph nodes that can be observed from node k by performing an observation action and $\mathcal{Q}_k^{t+\delta}$ is the list of nodes that are observed by the robot at the intermediate time step $t+\delta$ along its path from its current graph node n_r to node k. The reduction factor α is computed independently for each possible path.

We define $\mathcal{Q}_k^{t+\delta}$ such that it mainly considers the set of nodes that can be observed at that node without executing an observation action, i.e., a full rotation, by the robot. However, we additionally consider the possibility of performing observation actions at important locations which may lead to more information gain. During the HMM-based simulations we consider the intersection nodes at which performing an observation action might lead to a high information gain. We select such intersection nodes during the simulations based on the predicted belief of the user's presence at the graph nodes that can be observed only if an observation action is performed by the robot as follows:

$$\mathcal{Q}_k^{t+\delta} \leftarrow \begin{cases} \mathcal{O}_k^{t+\delta} & \text{if } \sum_{i \in \mathcal{O}_k^{t+\delta}} Bel_{t+\delta}(i) > c \ \sum_{i \in \mathcal{F}_k^{t+\delta}} Bel_{t+\delta}(i) \\ \mathcal{F}_k^{t+\delta} & \text{otherwise} \end{cases}, \tag{7}$$

where $\mathcal{F}_k^{t+\delta}$ is the set of nodes that can be observed *without* a rotation at the intermediate time step $t+\delta$ along the robot's path to current graph node to node k. Additionally, c is a constant factor that controls the selection threshold of these intersection nodes. This constant factor takes into account the additional time needed by the robot to perform a full rotation, such that a rotation is performed only when it leads to a net information gain.

The intersection nodes along the robot's path at which observation actions are useful are determined for each reachable node, i.e., while computing the observability likelihood of the user at each of them, according to Equation 7.

5.4. Selecting the Search Goal

After computing the observation likelihood l'_k for every node k, we select the node with the highest observability likelihood s as the next search location [1], i.e.,

$$s = \underset{k}{\operatorname{argmax}}\, l'_k. \tag{8}$$

The robot follows the shortest path in the graph to that search location and performs an observation action, i.e., a full rotation, when it is reached. If the robot cannot find the user along its way nor after performing the observation action, the search process is repeated by selecting a new search location as previously.

[1]Note that there is no need to append a time index to s since the time is implicitly considered in the computation of the observability likelihood.

As an example, Figure 4 shows the selection the next search goal as the location that is expected to provide highest observability along the robot's way to that location as well as at the time at which the robot reaches it.

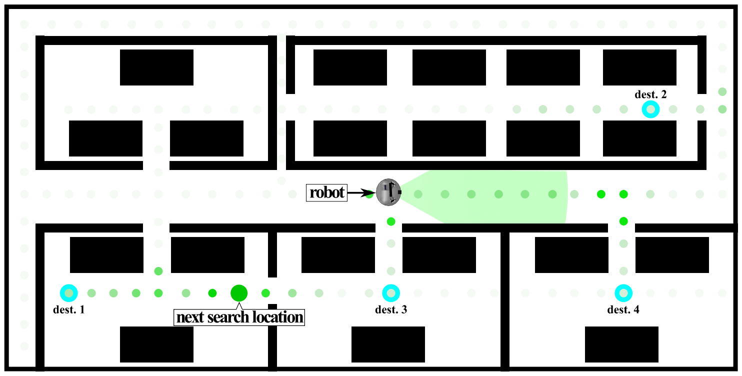

Figure 4: This figure shows the selected search location according to our approach. The graph nodes are drawn with a color intensity corresponding to the observability likelihood of the user at the time the robot reaches this search location. The robot selects the node that provides the highest observability likelihood as next search location.

6. Experimental Results

We evaluated our approach both in simulated and real-world experiments.

We carried out extensive experiments to evaluate our proposed framework, both in simulation and with a real robot, and compare it to alternative methods.

6.1. Experimental Setup

We performed the experiments in three different simulation environments (see Figure 2), 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. Additionally, we performed simulated experiments on a real indoor environment (as shown in Figure 3) of size 28.5 m $\times$ 9.7 m with a grid map resolution of 0.05 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 between the destinations are assumed to be equally likely. Note, however, that some 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. As can be seen, the visibility characteristics of the second environment are very difficult. In the third environment there exists only one path between any two nodes, for both the robot and the user. The fourth environment represents a real indoor environment with furniture and doors that increase the level of visual constraints. We manually generated path samples between all the possible destinations, where some of them were used for training the HMM and the rest was used for testing. Figure 5 and Figure 6 show the set of training paths.

Furthermore, we randomly generated the sequences of the destinations that the user visits during testing. Additionally, the sample paths used for testing include segments that deviate from the common paths that were used for training.

In each experiment, the position of the user is initialized according to the user's average occupation time for each graph node (see Section 4) and the user moves between the predefined destinations. The user does not necessarily move on the shortest path but might take detours. When the user reaches their destination, they wait there for a period of time sampled from a certain waiting interval. The user repeats this behavior until they reach their 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.

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. Furthermore, we set the constant factor c in Equation 7, which controls the selection of the intersection nodes, to 2.

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 $10\,\mathrm{m}$ view distance. We set

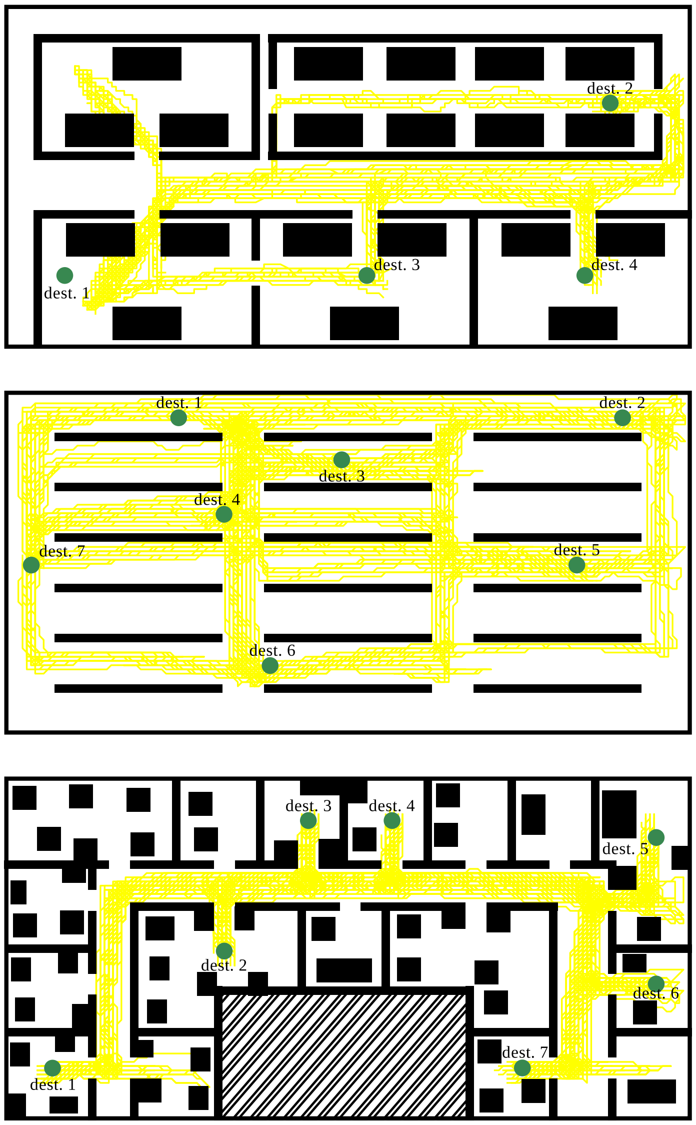

Figure 5: The randomly generated user's paths that are used for training overlaid on the three simulation environments.

the probability of false negatives between 0.05 and 0.15 linearly increasing with the distance between the robot and the user.

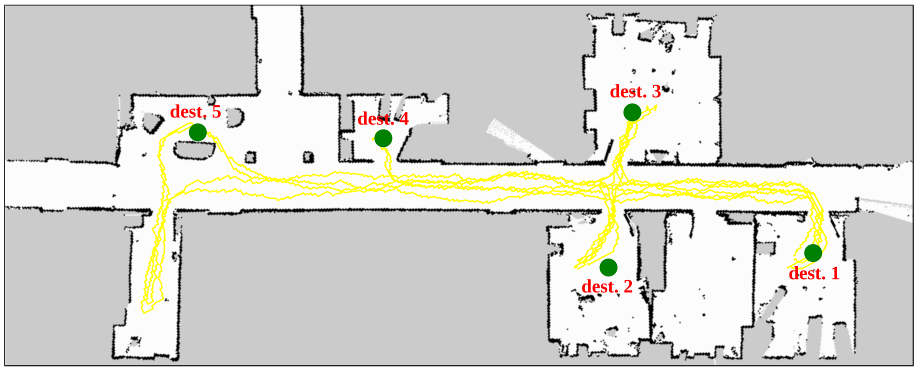

Figure 6: The randomly generated user's paths that are used for training overlaid on the real-world environment.

6.2. Results and Evaluation

We evaluated all three versions of our approach: The first version computes the user's observability likelihood relying only on the final node without considering observability at the intermediate nodes along the path (see Subsection 5.2) and we refer to this version as *HMM basic prediction*. The second version, which considers all extensions described in Subsection 5.3 is referred to it as *HMM with observability along the path*. The third version is the same as the second one except that it does not consider performing observation actions at important locations along the path and we refer to it as *HMM with observability along the path without rotations*. Furthermore, we compared the new HMM-based approaches to our previous work that relied on a particle filter [1] and we refer to it as *particle filter basic prediction*.

We performed $5,000$ experiments in each of the four environments where the robot's start location is randomly chosen at each experiment with the constraint that it does not observe the user from that initial location. We first compared our approach to a greedy maximum coverage approach that selects search locations which maximizes covering the so-far unexplored areas, i.e., it keeps selecting search locations that leads to covering the largest unexplored area considering both the areas that can be covered from that search location as well as the areas that can be covered along the path to such location from the robot's current location. After the search environment is totally covered, the strategy starts the search process all over again. We refer to this strategy as *greedy maximum coverage.*

Second, we compared the needed search time of all versions of our approach to the time needed by a greedy strategy that is fed with background information about the destinations of the user. The greedy strategy visits all destinations given information about the possible destinations of the user without considering any prediction about the user's location. It keeps selecting the closest unvisited destination as a search location until the user is found. After visiting all destinations it starts the search process all over again as in the case of the *greedy maximum coverage* strategy. We refer to this as *greedy with background information.*

Third, we compared our approach to a heuristic method that infers the user's most likely location in a step-by-step fashion using the hidden Markov model representation. Here, the robot moves toward that location for one time step, then it updates the estimation and so on. This method is similar to the approach of Goldhoorn *et al.* [19], although they use a particle filter instead of a HMM. We refer to this method as the *one-step to estimation* method.

We applied a *two-tailed paired t-test* to evaluate the statistical significance of our comparative experiments. The experimental results show that the three variants of our method performing HMM-based simulations significantly outperforms each of the two greedy approaches and the *one-step to estimation* strategy with a statistical significance of 99%.

Additionally, our approach *HMM with observability along the path* outperforms the *HMM basic prediction* and *particle filter basic prediction* approaches for all the environments with a statistical significance of 99%. Furthermore, our approach *HMM with observability along the path* outperforms its limited version *HMM with observability along the path without rotations* in the environments. This outperformance is achieved in the three simulation environments with a statistical significance of 99%. The *HMM with observability along the path* is slightly better than its limited version *HMM with observability along* *the path without rotations*, in the case of the real-world environment due to the limited points of intersection in such environment. This comparison highlights the importance of performing observation actions, i.e., rotations, at the important locations. Ignoring such rotations may lead to missing the user in some situations. Figure 7 shows the average relative search times achieved by all approaches normalized with respect to the greedy maximum coverage approach.

It can be clearly shown that the significance of our *HMM with observability along the path* approach is much higher in the case of the third environment, where a big number of isolated rooms exist. In such an environment, *greedy maximum coverage* suffers a lot by visiting all such rooms even if the user is not expected to visit all of them. The *greedy with background information* approach tends also to suffer due to the large number of scattered destinations. Furthermore, concerning the first two environments *HMM with observability along the path* approach could successfully deal with the multiple paths structure of this environment and this is highlighted especially in the second environment due to the massive number of intersections, where performing rotations at important intersections as determined by our approach leads to finding the user much faster. Concerning the fourth environment, which corresponds to a real world indoor environment, the search time of all the approaches is significantly lower due to the limited number of possible paths and according the limited number of intersections. Our *HMM with observability along the path* approach outperforms all other methods and achieves a lower search time.

Additionally, Figure 8 compares the distribution of the search times of the considered approaches. Note that this plot uses a log-scale at the y-axis. Thus, it can be shown clearly that the distributions of the different versions of our approach, especially our *HMM with observability along the path* approach, have a compact distribution which gives a more confidence about our results.

As all versions of our approach do not guarantee to completely cover the whole environment and, thus, might miss the user. On the other hand, the two greedy strategies, i.e., *greedy with background information* and *greedy maximum coverage*, are guaranteed to find the user because they keep starting their

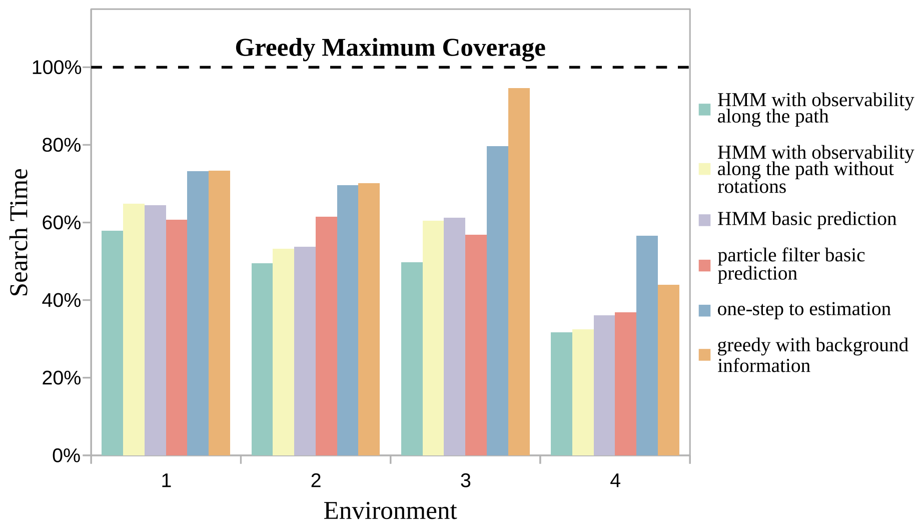

Figure 7: Average relative search time achieved by all versions of our HMM-based simulations approach compared to the other four approaches mentioned. The times are normalized so that the greedy maximum coverage approach equals 100%. The first three environments correspond to the environments shown in Figure 2, respectively, and the last one corresponds to the real-world environment shown in Figure 3. The three variants of our approach significantly outperform the *greedy maximum coverage strategy*, *greedy with background information* strategy and the *one-step to estimation* approach with a statistical significance of 99%.

process all over again when all the destinations are visited, or when the environment area is explored, respectively. Therefore, we switch to the *greedy with background information* approach after a given maximum time limit to avoid missing the user. This maximum time limit was determined experimentally such as to minimize the overall search time. We relied on the number of topometric graph nodes of each environment as a heuristic to determine this limit, since it gives insight about how many time steps is needed by the robot to visit the furthest graph node from its starting location. We started with this heuristic and kept tuning until we found the best values, which are 147, 175, 118 and 118 time steps for the four environments, respectively. Note that the chosen maximum time limit is the same for the compared strategies when performing the experiments.

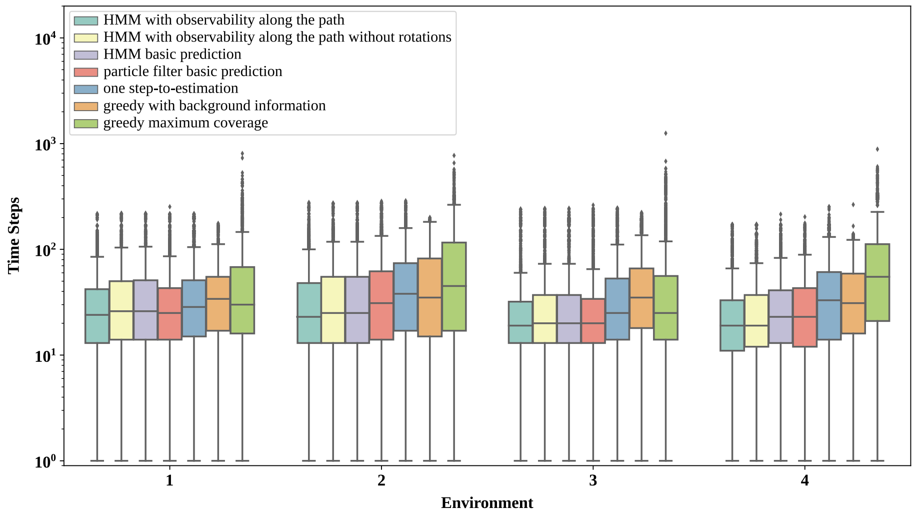

Figure 8: Whisker-box plot comparing the distribution of the search times (using log-scale) of each approach during the experimental runs on the three simulation environments and the real world environment. The first three environments correspond to the simulation environments shown in Figure 2, respectively, and the last one corresponds to the real-world environment shown in Figure 3.

Table 1 shows the percentage of experimental runs using our approaches exceeded this time limit and switched to the greedy method. We compare them to the *one-step to estimation* and the *particle filter basic prediction* approaches. Note that we included these additional search times resulting from switching to the greedy approach in all of the previously presented results.

As shown, our techniques based on HMM simulations outperform the *one-step to estimation* method for all the environments. Furthermore, our *HMM with observability along the path* approach outperforms all the other techniques on average. Although, the failure rate of the *particle filter basic prediction* approach is on average less than that of the *HMM basic prediction* approach, the *HMM basic prediction* approach is considered more robust because it does not depend on the random initialization and propagation of particles, i.e., re-running the experiments of the *particle filter basic prediction* multiple times leads to a varying performance due to the effect of the randomness. This variance in the

Table 1: Percentage of switching to the greedy with background information approach.

	Simulations-based Approaches				One-Step to Estimation
	HMM With Observability Along the Path		Basic Prediction		
	with Rotations	w/o Rotations	HMM	Particle Filter	
Env. 1	0.65%	2.29%	2%	1.87%	4.2%
Env. 2	1.35%	1.22%	1.4%	3%	4.3%
Env. 3	3.62%	7.39%	7.71%	5.7%	11.5%
Env. 4	2.64%	2.15%	2.54%	2.36%	17%
Over all Environments	2.07%	3.34%	3.5%	3.31%	9.02%

performance decreases as the number of particles increase. Accordingly, using a HMM which corresponds to a particle filter with an infinite number of particles will rule out this variance. This can be clearly figured out from the results in Table 1 and Figure 7 where the performance of *particle filter basic prediction* approach oscillates around that of the *HMM basic prediction* approach.

Two video showing the advantages of our approach can be downloaded from `https://www.hrl.uni-bonn.de/ras18bayoumi_sim_1.mp4` (simulation environments) and from `https://www.hrl.uni-bonn.de/ras18bayoumi_sim_2.mp4` (real-world environment).

6.3. Real-World Experiment

We also tested the practical applicability of our approach using a real robot. We deployed the Robotino robot from Festo equipped with an *ASUS Xtion Pro Live* camera and applied a particle filter for localization using laser range data. For user detection, we used an ArUco marker [20] as shown in Figure 9 (bottom left). The user spends most of his time at his office, i.e., *destination 1*, and he may move between the offices of his colleagues, i.e., *destination 2* and *destination 3*. Alternatively, he may grab a cup of coffee and drink it in the lobby, i.e, *destination 5*, or pick some printed documents from the printing room, i.e, *destination 4*.

For the experiment shown in Figure 9, the user is initially at his office, when the robot starts to search for it. Then, the user moves to *destination 2*.

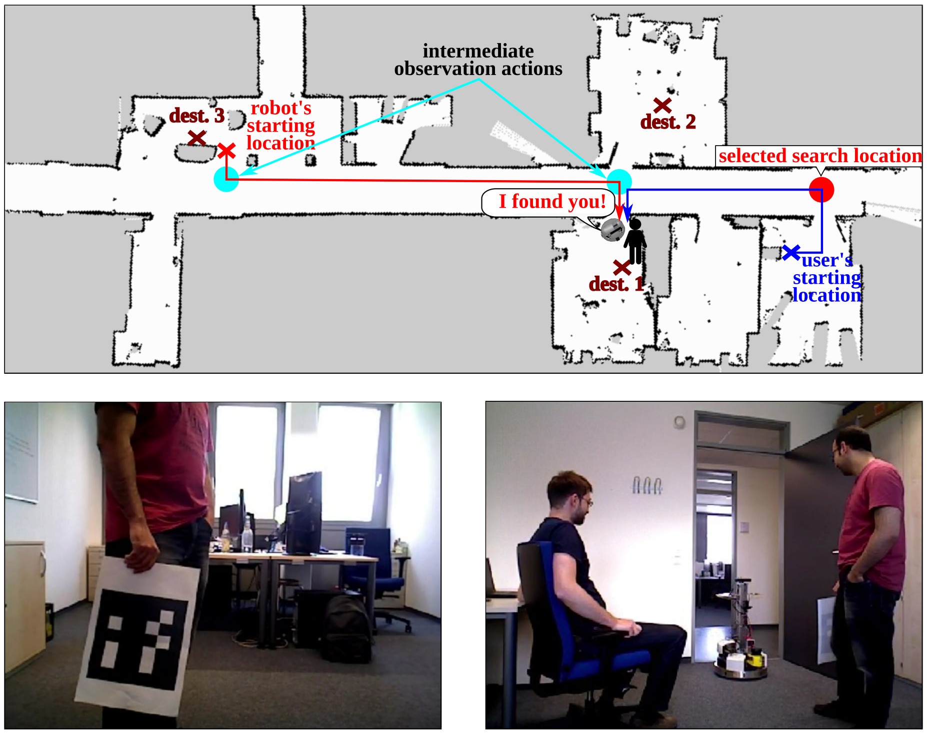

Figure 9: Top: Overview of our real-world environment showing both the robot's and the user's paths (in red and blue, respectively), the initially selected search location, as well as their final locations. The user's possible destinations are shown in maroon. The robot drives along the shortest path to the selected search location, additionally, it performs intermediate observation actions at the highlighted cyan locations along its path. Bottom: The robot has found the user after performing an observation action.

The robot selects a search location near to the user's office while considering performing observation actions at important locations along the robot's path. The robot successfully finds the user while performing one of those intermediate observation actions. A video showing this experiment can be downloaded from `https://www.hrl.uni-bonn.de/ras18bayoumi_real_world.mp4`.

7. Conclusions

In this article, we presented an approach that enables a mobile robot to efficiently find a non-stationary user in complex environments. Our method selects the best next search goal by simulating future paths of the user. To compute the likelihood of observing of the user, we perform simulations using a hidden Markov model given a graph representation of their known, typical paths in the environment. To select a good search location and observation actions, we consider the travel time needed by the robot to reach the search locations and visibility constraints.

As our simulation and real-world experiments demonstrate, our approach enables the robot to find the user within a short amount of time by selecting effective search locations. We showed in extensive experiments that our framework outperforms two other common search methods.

Acknowledgments

This work has been supported by the German Academic Exchange Service (DAAD) and the Egyptian Ministry for Higher Education and by the DFG Research Unit FOR 2535 Anticipating Human Behavior.

References

[1] A. Bayoumi, P. Karkowski, M. Bennewitz, People finding under visibility constraints using graph-based motion prediction, in: Proc. of the Int. Conf. on Intelligent Autonomous Systems (IAS), 2018.

[2] I. Suzuki, M. Yamashita, Searching for a mobile intruder in a polygonal region, SIAM Journal on Computing 21 (5) (1992) 863–888.

[3] L. Guibas, J. Latombe, S. LaValle, D. Lin, R. Motwani, A visibility-based pursuit-evasion problem, International Journal of Computational Geometry and Applications (1996) 471–494.

[4] J.-H. Lee, S.-M. Park, K.-Y. Chwa, Searching a polygonal room with one door by a 1-searcher, International Journal of Computational Geometry & Applications 10 (02) (2000) 201–220.

[5] V. Isler, S. Kannan, S. Khanna, Randomized pursuit-evasion in a polygonal environment, IEEE Transactions on Robotics 21 (5) (2005) 875–884.

[6] H. Choset, Coverage for robotics – a survey of recent results, Annals of Mathematics and Artificial Intelligence 31 (1) (2001) 113–126.

[7] N. Stiffler, A. Kolling, J. O'Kane, Persistent pursuit-evasion: the case of preoccupied pursuer, in: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), 2017, pp. 5027–5034.

[8] M. Moors, T. Röhling, D. Schulz, A probabilistic approach to coordinated multi-robot indoor surveillance, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), 2005, pp. 3447–3452.

[9] B. Gerkey, S. Thrun, G. Gordon, Visibility-based pursuit-evasion with limited field of view, Int. Journal of Robotics Research (IJRR) 25 (4) (2006) 299–315.

[10] A. Kolling, S. Carpin, Multi-robot surveillance: An improved algorithm for the graph-clear problem, in: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), 2008, pp. 2360–2365.

[11] G. Hollinger, S. Singh, A. Kehagias, Improving the efficiency of clearing with multi-agent teams, Int. Journal of Robotics Research (IJRR) 29 (8) (2010) 1088–1105.

[12] G. Tipaldi, K. Arras, 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, pp. 1217–1222.

[13] M. Schwenk, T. Vaquero, G. Nejat, K. Arras, Schedule-based robotic search for multiple residents in a retirement home environment, in: Proc. of the National Conf. on Artificial Intelligence (AAAI), 2014, pp. 2571–2577.

[14] M. Kulich, T. Krajník, L. Přeučil, T. Duckett, To explore or to exploit? Learning humans' behaviour to maximize interactions with them,

in: J. Hodicky (Ed.), Modelling and Simulation for Autonomous Systems: Third International Workshop, MESAS 2016, Revised Selected Papers, Springer International Publishing, 2016, pp. 48–63.

[15] T. Krajník, M. Kulich, L. Mudrová, R. Ambrus, T. Duckett, 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, pp. 2140–2146.

[16] M. Volkhardt, S. Mueller, C. Schroeter, H.-M. Gross, Playing hide and seek with a mobile companion robot, in: Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids), 2011, pp. 40–46.

[17] M. Volkhardt, H.-M. Gross, Finding people in home environments with a mobile robot, in: Proc. of the European Conf. on Mobile Robots (ECMR), 2013, pp. 282–287.

[18] S. Mehdi, K. Berns, Behavior-based search of human by an autonomous indoor mobile robot in simulation, Universal Access in the Information Society 13 (1) (2014) 45–58.

[19] A. Goldhoorn, A. Garrell, R. Alquézar, A. Sanfeliu, Searching and tracking people in urban environments with static and dynamic obstacles, Robotics & Autonomous Systems 98 (2017) 147–157.

[20] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, M. Marín-Jiménez, Automatic generation and detection of highly reliable fiducial markers under occlusion, Pattern Recognition 47 (6) (2014) 2280 – 2292.