Where's Waldo at time t ?
Using Spatio-Temporal Models for Mobile Robot Search

Tomáš Krajník[1] Miroslav Kulich[2] Lenka Mudrová[3] Rares Ambrus[4] Tom Duckett[1]

***Abstract*— We present a novel approach to mobile robot search for non-stationary objects in partially known environments. We formulate the search as a path planning problem in an environment where the probability of object occurrences at particular locations is a function of time. We propose to explicitly model the dynamics of the object occurrences by their frequency spectra. Using this spectral model, our path planning algorithm can construct plans that reflect the likelihoods of object locations at the time the search is performed.**

Three datasets collected over several months containing person and object occurrences in residential and office environments were chosen to evaluate the approach. Several types of spatio-temporal models were created for each of these datasets and the efficiency of the search method was assessed by measuring the time it took to locate a particular object. The results indicate that modeling the dynamics of object occurrences reduces the search time by 25% to 65% compared to maps that neglect these dynamics.

***Index Terms*— mobile robotics, long-term autonomy**

I. Introduction

Searching for an object is a practical task in everyday life that would be trivial if we had perfect knowledge about our own environment. Knowing the position of a searched object in advance means that search does not have to be performed at all. Unlike the highly structured and well-defined worlds of factory assembly lines, which are engineered to minimise the uncertainty in object positions, the locations of most objects in human-populated environments are inherently uncertain. The uncertainty of human-populated environments is induced by the humans themselves, as they change not only their own locations, but also the positions of the objects they use. This motivates the need to understand (or at least represent) the dynamic processes that influence the locations of the humans and the items that they use.

We assume that the search task is supposed to be carried out by a mobile robot that is able to detect the searched object using its on-board sensors. To perform the search efficiently, the robot has to be able to plan its motion through the environment. Better knowledge about possible object location leads to more efficient plans and hence, shorter times to locate the desired object. Most path-planning algorithms require this knowledge in the form of a probability distribution over possible object locations. This distribution is typically constructed by means of probabilistic inference that takes into account several factors including the history of the searched object locations.

We argue that in human-populated environments, the object locations are primarily influenced by human activities that tend to exhibit daily and weekly routines. We show that identification and modeling of these routines leads to a more faithful representation of possible object locations and hence to a faster and more efficient search. To represent the probabilities of object locations we use a traditional topological map where each node is associated with a temporal model that captures the dynamics of the object occurrence at that particular location. These temporal models are based on Periodic Gaussian Mixtures similar to the approach proposed in [1] and the Frequency Map Enhancement (FreMEn) technique [2] that allows to introduce dynamics into static environment models. We evaluate the Gaussian- and spectral-based temporal models on datasets gathered over several weeks and show that the search performed by a mobile robot is faster when using spatio-temporal models with probabilities that reflect the environment dynamics.

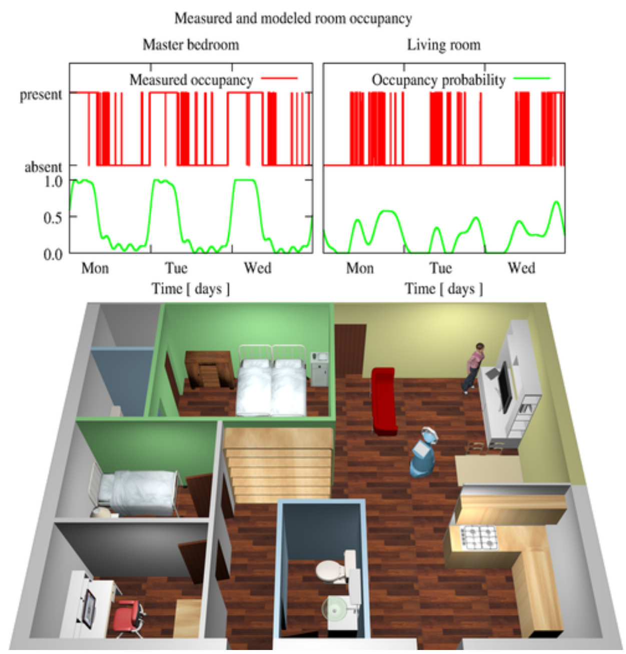

Fig. 1: Example environment with probabilities of person presence in two rooms.

[1]Lincoln Centre for Autonomous Systems, University of Lincoln, UK {tkrajnik,duckett}@lincoln.ac.uk

[2]Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic {kulich}@fel.cvut.cz

[3]Intelligent Robotics Lab, University of Birmingham, UK {lxm210}@cs.bham.ac.uk

[4]KTH - Royal Institute of Technology, Sweden {raambrus}@kth.se

II. Related work

The problem of finding a static object of interest has been studied from various perspectives. The operational research community defines the Minimum Latency problem (MLP, also known as the Traveling Deliveryman Problem or the Traveling Repairman Problem) as a problem of constructing a tour through all nodes in a graph that minimizes the sum of latencies of the nodes, where the latency of a node is the distance needed to travel to that node. The problem has been proven to be NP-hard [3]. Nevertheless several exact exponential time algorithms were introduced such as an integer linear programming approach [4] and a branch and bound algorithm [5]. Moreover, several approximate algorithms were presented recently. Salehipour et al. [6] present a meta-heuristic combining General Randomized Adaptive Search (GRASP) with Variable Neighborhood Descent (VNS). A meta-heuristic called GVNS (General Variable Neighborhood Search) is introduced in [7], while integer linear programming is used in [8].

Koutsoupias et al. [9] extend MLP by introducing the Graph Searching Problem (GSP), where the latency of each node is multiplied by a constant weight associated with that node, and prove that GSP can be reduced to MLP under certain conditions. Furthermore, Ausiello et al. [10] present a reduction algorithm for the metric GSP and the GSP on tree networks.

While the previous approaches are defined on a graph, Sarmiento et al. [11] formulate the problem in the polygonal domain, where the time required to find a static object is a random variable induced by the choice of search path and a uniform probability density function over the object's location. They propose a two-stage process to solve the problem: a set of locations (known as 'guards' from the art gallery problem [12]) to be visited is determined first, followed by finding the order of visiting those locations to minimize the expected time to find an object. The optimal order is determined by a greedy algorithm in a reduced search space, which computes a utility function for several steps ahead. This approach is then used in [13], where robot control is assumed in order to generate smooth and locally optimal trajectories. Recently, Kulich et al. presented the search problem for a robot operating in an unknown environment [14].

Other approaches model spatial relations about objects in the scene and use these relations to constrain the space of possible object locations. It has been shown that the refinement of the locations considered for search has a positive influence on search performance [15]. For example, object search based on Qualitative Spatial Relations is presented in [16], while five priors representing structure of the world and the specific scene are encoded into a probabilistic model and used to build consistent hypotheses about object locations in [17].

The idea of modeling the long-term dynamics of indoor environments was presented in [2], where the authors argue that part of the environment variations exhibit periodicities and represent the environment states by their frequency spectra. The concept of Frequency-based Map Enhancement (FreMEn) was applied to occupancy grids in [18] to achieve compression of the observed environment variations and to landmark-based maps in order to increase robustness of mobile robot localization [19]. In this paper, we show that usage of dynamic maps based on the same concept allows for better planning in the context of mobile robotic search.

III. Problem formulation

The search problem can be generally understood as navigation through an environment in order to find an object with an unknown location. By finding an object we mean the situation when it is first detected by the robot's sensors. A criterion to minimize is the time when this situation occurs. In this paper, we assume that the topology of the environment is known a priori and the object to be found remains stationary during the search.

The problem is formally defined as a variant of the Graph Searching Problem [9]. That is, given

- an undirected graph $G(V, E)$, where V is a finite set of vertices and E is a set of edges between these vertices,
- $d : \; E \to \mathbb{R}$ the time needed to traverse the edge, and
- $p \; : \; V \to \langle 0, 1 \rangle$ the probability of presence of the searched object at the given vertex,

the objective is to find a walk $\boldsymbol{\omega}^* = \langle \omega_0, \omega_1, \dots \omega_k \rangle$ in G, which visits all nodes of V at least once (i.e. $\forall v \in V \; \exists \omega_j \in \boldsymbol{\omega}^* : \; \omega_j = v$) and which minimizes the expected time to find the object as

$$\boldsymbol{\omega}^* = \arg \min_{\boldsymbol{\omega} \in \Omega} \mathbb{E}(T|\boldsymbol{\omega}), \tag{1}$$

where Ω is a set of all possible walks in G and

$$\mathbb{E}(T|\boldsymbol{\omega}) = \sum_{i=1}^{|\boldsymbol{\omega}|} \left(p(\omega_i) \sum_{j=1}^{i} d(\omega_{j-1}, \omega_j) \right). \tag{2}$$

The minimal expected time is then

$$T_{exp} = \mathbb{E}(T|\boldsymbol{\omega}^*). \tag{3}$$

IV. Temporal models

The underlying environment representation used in our approach is a topological map, where nodes represent distinct areas and edges represent the robot's ability to move between them. Each node of our map is associated with a probability of person presence and each edge contains information about the time it takes to move between the map's nodes. Unlike traditional maps, the probabilities associated with particular nodes are not constant, but are functions of time. These functions are estimated through long-term observation of the person or object presence at the given locations.

A. *Frequency map enhancement*

Frequency Map Enhancement (FreMEn) is a method that can introduce dynamics into static environment models [2]. So far it was applied to mobile robotic mapping and localization in scenarios where the robots are required to operate autonomously for long periods of time. FreMEn assumes that most of the indoor environment states are influenced by humans that perform their regular daily activities. The regularity and influence of these activities on the environment states is obtained by means of frequency transforms. Specifically, FreMEn extracts the frequency spectra of binary functions that represent long-term observations of environment states, discards non-essential components of these spectra and uses the remaining spectral components to represent probabilities of the corresponding binary states in time. The authors of FreMEn have shown that introducing dynamics into volumetric, topological and landmark-based environment models enables these to represent the world more faithfully [2], [18], which in turn results in increased robustness of mobile robot self-localization [19].

Let us assume that the presence of an object at a particular area of the environment is represented by a binary function of time $s(t)$. Let us represent the uncertainty of the state $s(t)$ by its probability $p(t)$. Then, the main idea of FreMEn is to represent a (temporal) sequence of states $s(t)$ by the most prominent components $P(\omega)$ of its frequency spectrum $S(\omega)$ = $\mathcal{F}(s(t))$, where $\mathcal{F}(.)$ represents the Fourier transform. The advantage of this representation is that each spectral component of $P(\omega) \subset S(\omega)$ is represented by three numbers only, which leads to high compression rates of the observed sequence $s(t)$.

To create the FreMEn model, the frequency spectrum $S(\omega)$ of the sequence $s(t)$ is first calculated. The first spectral component a_0, which represents the average value of $s(t)$ is stored. The remaining spectral components of $S(\omega)$ are ordered according to their absolute value and the n highest components are selected. Each component represents a harmonic function that is described by three parameters: amplitude a_j, phase shift φ_j and frequency ω_j. The superposition of these components, i.e.

$$p(t) = a_0 + \sum_{j=1}^{n} a_j cos(\omega_j t + \varphi_j), \tag{4}$$

allows to estimate the probability $p(t)$ of the state $s(t)$ for any given time t. Since t is not limited to the interval when $s(t)$ was actually measured, Eq. (4) can be used not only to interpolate, but also to predict the state of a particular model component. In our case, we use Eq. (4) to predict the presence of the searched object in a particular room.

While the aforementioned representation deals well with periodicities, it suffers from two disadvantages:

- it allows to model only one process per frequency,
- it poorly models regular, but short-duration events.

Many of the daily activities that occur in an apartment or an office exhibit both of the aforementioned characteristics. Examples include preparing a hot drink, brushing teeth or taking a shower – these activities typically occur on a regular basis and take a short time. Moreover, some of the aforementioned activities influence the presence of objects in the same room.

Application of the Fourier Transform to temporal sequences that possess the aforementioned characteristics causes it to approximate these functions with high-frequency harmonic components. However, these components do not correspond to any regular patterns that actually occur in the environment and therefore decrease the predictive capabilities of the frequency-based temporal model. The results presented in [2], [19] suggest that the best predictive capabilities are achieved by FreMEn models of the 2^{nd} or 3^{rd} order. Such low-order models are not able to represent short-duration events.

B. Gaussian Mixture Models

Gaussian Mixture Models that can approximate multi-dimensional functions as a weighted sum of Gaussian component densities are a well-established method of function approximation. A Gaussian Mixture Model of a function $f(t)$ is a weighted sum of m Gaussian functions:

$$f(t) = \frac{1}{\sqrt{2\pi}} \sum_{j=1}^{m} \frac{w_j}{\sigma_j} e^{-\frac{(t-\mu_j)^2}{2\sigma_j^2}}. \tag{5}$$

GMMs find their applications in numerous fields ranging from Botany to Psychology [20]. The parameters of individual components of GMMs, i.e. the weights w_k, means μ_j and variances σ_j are typically estimated from training data using the iterative Expectation Maximization (EM) or Maximum A-Posteriori (MAP) algorithms. While GMMs can model arbitrarily-shaped functions, their limitation rests in the fact that they cannot naturally represent functions that are periodic.

To deal with this issue, we simply assume that people perform most of their activities on a daily basis and thus we consider the object presence in the individual areas as being the same for every day. While this assumption is not entirely correct (as working days will be different from weekends), such a temporal model might still be better than a 'static' model where the probability of object presence is a constant.

Prior knowledge of the periodicity allows to transform the measured sequence of states $s(t)$ into a sequence $p'(t)$ by

$$p'(t) = \frac{k}{\tau} \sum_{i=1}^{k/\tau} s(t + i\tau), \tag{6}$$

where τ is the assumed period and k is the $s(t)$ sequence length. After calculating $p'(t)$, we employ the Expectation Maximization algorithm to find the means μ_j, variances σ_j and weights w_j of its Gaussian Mixture approximation. Thus, the probability of occupancy of a room at time t is given by

$$p(t) = \frac{1}{\sqrt{2\pi}} \sum_{j=1}^{m} \frac{w_j}{\sigma_j} e^{-\frac{(mod(t,\tau)-\mu_j)^2}{2\sigma_j^2}}, \tag{7}$$

where τ is the a priori known period of the function $p(t)$ and mod is a modulo operator. The advantages of this periodic-GMM-based (PerGaM) model are complementary to the weaknesses of the FFT-based one. It can approximate even short, multiple events, but it can represent only one period that has to be known a priori.

An example comparison of the PerGaM and FreMEn models of person presence in a week-long experiment in an office environment is shown in Figure 2. The figure

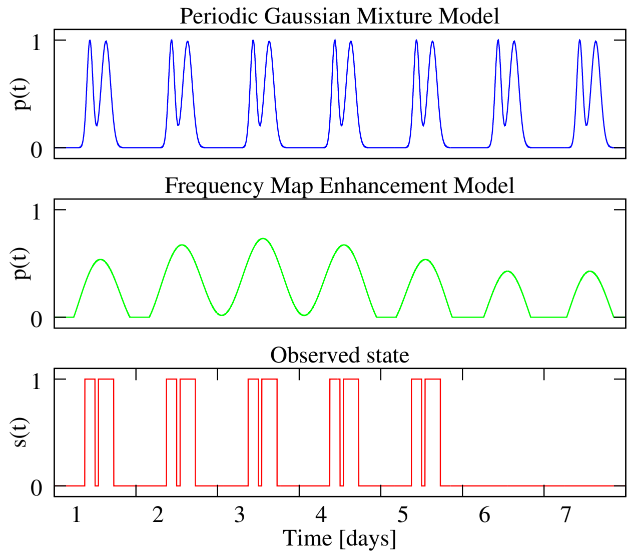

Fig. 2: PerGaM and FreMEn models example/comparison.

demonstrates that while PerGaM can model short-term event like lunch-breaks, it fails to capture the week-long dynamics.

V. Search Algorithm

The proposed path-planning search algorithm is a variant of branch-and-bound based on a recursive version of depth first search (DFS).

Procedure BranchAndBound(walk)

```
1  if all vertices visited then
2      if current walk faster than the fastest one then
3          set the current walk as the fastest one;
4  else if estimated walk time exceeds fastest walk then
5      return
6  else
7      foreach not visited vertex do
8          backup the current state;
9          append the shortest path P to the last vertex;
10         update the cost (duration) of walk;
11         mark all vertices in P as visited;
12         call BranchAndBound(walk);
13         restore the state;
```

The algorithm systematically constructs all possible walks (see **Procedure** BranchAndBound) in G by calling BranchAndBound($start$), where $start$ is the vertex where the robot starts searching. The procedure starts by testing whether the current walk contains all vertices of G. If this is the case and the walk w is faster than the current fastest solution $fastest$ (i.e., $\mathbb{E}(T|w) < \mathbb{E}(T|fastest)$) then w becomes the new fastest solution (lines 2-3). If w is worse than $fastest$ the procedure is finished, leading to backtracking of the BFS (line 5). Remaining vertices to visit are sequentially processed at lines 7-13. For each unvisited vertex v, the shortest (in time) path P is determined from the last vertex in w to v and appended to w: $w = w \cup P$ (line 9). Moreover, the new cost of w according to Eq.(3) is computed and all vertices in P are labeled as visited (lines 10-11). BranchAndBound is then called on the new walk at line 12. Note that backing up and restoring of the current state (walk, its cost, and the marking of visited vertices) at lines 8 and 13 allows for efficient backtracking.

To achieve effective branching and thus low computation times, several improvements to the original algorithm were made. Firstly, the vertices in the *foreach* cycle at line 7 are processed in increasing distance to the last vertex in w, i.e. more promising shorter edges are processed first. Secondly, Johnson's algorithm [21] is run in the preprocessing phase to find the shortest paths between all pairs of vertices, which are used at line 9. Finally, the estimated cost (i.e. the lower bound of all walks with a prefix w) $T^{exp}(w)$ at line 4 is determined as

$$
\begin{aligned}
T^{exp}(w) &= \mathbb{E}(T|w) + \sum_{i=0}^{K} p_i D_i, \text{where} \\
D_i &= D(w) + \sum_{\iota=0}^{i} d_\iota,
\end{aligned}
$$

where K is the number of unvisited vertices, $D(w)$ is the length of w, and d is an array containing the shortest edges incident to the unvisited vertices sorted in ascending order. Similarly, p is a descendingly sorted array of weights (presence probabilities) of the unvisited vertices.

The algorithm is exact and solves small-size instances in a reasonable time, i.e. the 'Brayford' problem with 11 vertices (see section VI) in less than 1 ms, a graph with 20 vertices in 6 sec and a graph with 25 vertices in 60 sec on a standard computer.

VI. Experiments

To evaluate the utility of temporal models to speed-up the robotic search, we performed an extensive comparison of the proposed methods on two long-term datasets where the searched object is a human and one dataset with several objects in everyday use. Each dataset was divided into separate training and testing sets. The training sets were used to establish the temporal models of the nodes' occupancy (person or object presence) and edge traversals times. The testing sets were used to establish the time statistics of the search.

The first, 'Aruba' dataset [22] was collected by the Center for Advanced Studies in Adaptive Systems (CASAS) to support their research concerning smart environments. The second 'Brayford' dataset [19] was created by the Lincoln Centre for Autonomous System (LCAS) as a part of the collaborative EU-funded STRANDS project, which aims to enable long-term autonomous operation of intelligent robots in human-populated environments. The third 'KTH' dataset was created by the Royal Institute of Technology in Stockholm, also as part of the STRANDS project.

A. Aruba dataset

The 'Aruba' dataset contains measurements collected by 50 different sensors distributed over a 12×10 m, seven-room apartment over a period of 16 weeks. The apartment is occupied by a single person who is occasionally visited by other people. To estimate person presence in the individual rooms of the apartment, we processed the ON/OFF events from the dataset's motion detectors. A room is considered to be occupied if any of its motion sensors report movement. In case no motion is detected by any of the sensors, the occupancy value of the corresponding room is unchanged, i.e. the last room reported to be occupied retains its state.

The apartment was partitioned into nine different areas, seven of which represent the rooms and two correspond to corridors – see Figures 1 and 3. Thus, we obtained nine sequences that represent the occupancies of individual areas second-by-second for 16 weeks. The first 4 weeks of the data were used to train the temporal models of room occupancy and the remaining 12 weeks of the data were used as a testing set, i.e. for the evaluation itself. Thus, each areas' occupancy training and testing set consists of more than 2.4 million and 7.2 million samples, respectively. Note that the occupancy probabilities of the individual areas do not have to sum to one, because the apartment is sometimes completely empty and the inhabitant is occasionally visited by relatives. To

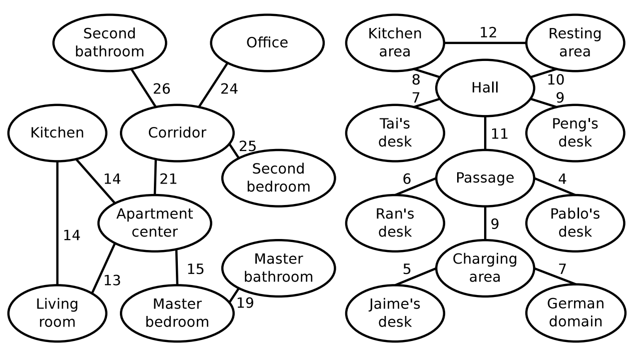

Fig. 3: Aruba and Brayford datasets - topological maps.

establish the times it takes to move between individual areas, a simulated environment based on the 'Aruba' apartment layout was created, see Figure 1. A virtual SCITOS-G5 mobile robot was placed in the apartment model and set up to autonomously navigate between the individual rooms. The robot's trajectory was recorded, and the average times it took to traverse between the individual areas were calculated and used as edge traversal times of the topological map used by the planning algorithm. The average times (in seconds) are shown on the Aruba topological map, see Figure 3. Thus, the resulting topological representation is based on both real-world data (room occupancies) and simulated data (time to navigate between rooms).

B. Brayford dataset

The Brayford dataset was collected by a SCITOS-G5 mobile robot (see Figure 4) equipped with an RGB-D camera mounted on a pan-tilt unit and a laser rangefinder. The robot was programmed to patrol a large, open-plan office of the Lincoln Centre for Autonomous Systems. Its autonomous navigation was based on an improved ROS navigation stack and a visual-based method for precise docking at the robot's charging station [23]. While the robot's laser rangefinder was used for autonomous navigation, the robot's Asus Xtion RGB-D camera was used for data collection. The robot was

Fig. 4: The SCITOS-G5 robot and example images of the Brayford dataset.

set-up to capture RGB images of eight designated areas of the office every ten minutes for two weeks of November 2013 and one week of February 2014. While the first week of the November data was used as a training set, the testing set consisted of two days – one captured during November 2013 and one on February 2014. The training and testing datasets consisted of approximately 10000 images that were manually checked for people presence (see Figure 4). To establish the edge traversal times, the trajectory recorded during data collection was analysed similarly to the Aruba dataset.

C. KTH dataset

One might argue that the Aruba and Brayford datasets are not related to object search since the robot searches for people. To verify whether the locations of objects in everyday use also exhibit periodicities, a specialized dataset was created at another site. The KTH dataset was collected by a SCITOS-G5 mobile robot (see Figure 4), in the Computer Vision and Active Perception lab at KTH Stockholm, over the course of five weeks. During this time the robot conducted between two and six autonomous patrol runs per day (weekends were excluded), visiting three specific waypoints during each run. Upon reaching a waypoint, the robot would execute a pan-tilt sweep and collect data from its RGB-D sensor; the RGB-D frames collected during one sweep were then registered spatially to form an observation of that particular waypoint at that time. The KTH dataset contains approximately 100 observations per waypoint, and at each waypoint we extracted the dynamic elements of the environment using the 'MetaRoom' method described in [24].

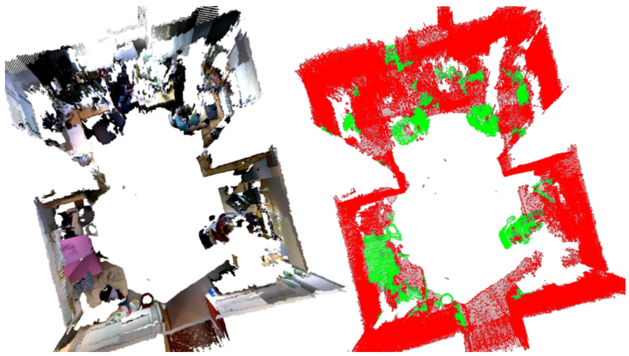

Fig. 5: Example observations with identified static (red) and dynamic (green) structures.

These dynamic elements correspond to movable objects such as jackets, backpacks, laptops, chairs, bottles, mugs, etc. For the experiments presented in this paper, we manually labeled these dynamic clusters to obtain 37 different objects, out of which 14 tend to appear and disappear periodically. Using the first four weeks of the KTH dataset, dynamic models of these objects' presence were created. The remaining week was used as testing data.

VII. Results

A. Aruba and Brayford datasets

To evaluate the influence of the temporal models on the time taken for the robotic search, we first built seven temporal models of each of the two environments and used these as an input for the search algorithm described in Section V. Three temporal models were based on the FreMEn concept and differed in the number (one to three) of spectral components included in the model. Another three 'PerGaM' temporal models consisted of the Periodic Gaussian Mixtures, where the daily occupancy was modeled by a combination of one-to-three Gaussians. Finally, a reference, 'Static' model represented the occupancy of each area by a constant probability. Each of these models was used to predict the probabilities of area occupancies for every minute of the Aruba and every 10 minutes of the Brayford testing sets. Each of the topological maps generated was used as an input for the planning algorithm V that calculated an optimal (in the sense of criterion (1)) robot path. Using the ground truth data of the testing sets, we determined the time it would take for the robot to find a person at a particular time using a given temporal model. The testing set of the Aruba environment allowed us to perform over ∼850 000 search runs (7 evaluations every minute for 12 weeks) and the Brayford results are based on ∼2000 testing runs (7 evaluations every 10 minutes for two days). To indicate the impact of using the temporal models for the robot search, we

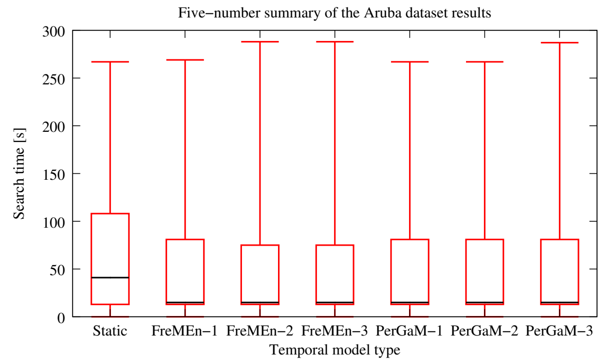

Fig. 6: Five-point summary of search times for different temporal models - Aruba dataset.

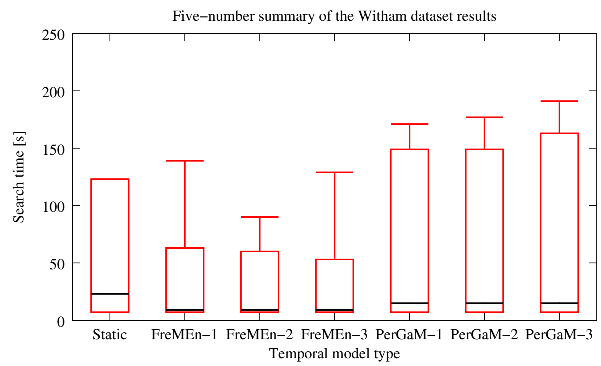

Fig. 7: Five-point summary of search times for different temporal models - Brayford dataset.

provide graphs of five-point characteristics extracted from the aforementioned testing runs. Figures 6 and 7 show that in general, usage of the temporal models allows to construct plans that are more likely to find a person in a given environment faster than in cases when the temporal information is neglected (the 'Static' models). Compared to the stationary models, the *median* time of the dynamic models' testing runs is lower by 35-65%, while the *average* search time was reduced by approximately 25%. However, Figure 6 also indicates that in about 6% of cases, the plan that takes into account the temporal models performs worse than the one using a static model. This is typically caused by short-term absences of persons at the expected locations and cases when the environment is completely vacant.

To further evaluate the results, we calculated the mean times it takes to find the person when the environment is not empty. The mean time of finding a person along with the median time to complete the search is shown in Table I.

B. KTH dataset

The KTH dataset is different from the Aruba and Brayford ones, because the objects sometimes appear at a particular waypoint only. Instead of having the robot plan a sequence of rooms to move through, we decided it should find a set of objects at this waypoint and let it decide when to visit that waypoint. Instead of visiting the place at regular

TABLE I: Mean (μ) and median ($\tilde{t}$) time to find a person

Model		Dataset			
		Aruba		Brayford	
type	order	$\mu[s]$	$\tilde{t}[s]$	$\mu[s]$	$\tilde{t}[s]$
Static	-	44	41	19	23
FreMEn	1	36	15	14	9
FreMEn	2	33	15	14	9
FreMEn	3	34	15	16	9
PerGaM	1	34	15	14	15
PerGaM	2	33	15	14	15
PerGaM	3	33	15	14	15

intervals, the robot calculates the joint probability of the objects' presence for the time of the intended patrol and decides to visit the place only when the probability of the searched objects exceeds 0.5. This experiment shows that this policy decreased the number of visits needed to find all the objects by 33%.

VIII. Conclusion

We have presented a novel approach to the mobile robot search problem based on spatio-temporal environment modeling and efficient probabilistic path planning. We assume that the topology of the robot's operational environment is known and that the searched object locations are influenced by human activities which tend to exhibit a certain degree of periodicity. This assumption allows to model the likelihood of object occurrences at particular locations as a combination of periodic functions which are identified from the long-term observations of object occurrences. These functions constitute dynamic probability distributions of the searched objects' locations, which allow to create different search plans for different times of day. In other words, our approach allows to integrate several observations of the same environment in a spatio-temporal model that captures the periodic aspects of object occurrences and uses this knowledge to construct time-dependent plans for the object search.

The experimental evaluation performed on datasets gathered over several weeks show that explicit representation of the long-term periodicities of environment dynamics speeds up the search process. Compared to traditional probabilistic models that neglect long-term environment dynamics, using the proposed spatio-temporal models for path planning resulted in a 25% - 65% reduction in the time needed to locate the searched object.

The tests have shown that the spectral-based dynamic models speed up the search slightly more than the Gaussian-based ones, though the difference was not statistically significant in our experiments. Since these two temporal models have complementary strengths and weaknesses, we plan to combine them in a two-stage method that first identifies the periodicities by a Fourier Transform and then approximates the periodic events by Gaussian Mixtures. We also plan to experimentally verify other temporal models, such as Gaussian processes.

Acknowledgments

The work was funded by the EU ICT project 600623 'STRANDS' and the Technology Agency of the Czech Republic project TE01020197 'CAK'.

References

[1] N. Haji Ghassemi and M. Deisenroth, "Analytic long-term forecasting with periodic Gaussian processes," in *Seventeenth International Conference on Artificial Intelligence and Statistics*, 2014, pp. 303–311.

[2] T. Krajník, J. P. Fentanes, G. Cielniak, C. Dondrup, and T. Duckett, "Spectral analysis for long-term robotic mapping," in *International Conference on Robotics and Automation (ICRA)*, 2014.

[3] F. Afrati, S. Cosmadakis, C. H. Papadimitriou, G. Papageorgiou, and N. Papakostantinou, "The complexity of the travelling repairman problem," *Theoretical Informatics and Applications*, 1986.

[4] M. Fischetti, G. Laporte, and S. Martello, "The delivery man problem and cumulative matroids," *Operations Research*, pp. 1055–1064, 1993.

[5] B. Y. Wu, Z.-N. Huang, and F.-J. Zhan, "Exact algorithms for the minimum latency problem," *Information Processing Letters*, 2004.

[6] A. Salehipour, K. Sörensen, P. Goos, and O. Bräysy, "Efficient GRASP+VND and GRASP+VNS metaheuristics for the traveling repairman problem," *4OR*, vol. 9, pp. 189–209, 2011.

[7] N. Mladenović, D. Urošević, and S. Hanafi, "Variable neighborhood search for the travelling deliveryman problem," *4OR*, pp. 1–17, 2012.

[8] I. Méndez-Díaz, P. Zabala, and A. Lucena, "A new formulation for the traveling deliveryman problem," *Discrete Applied Mathematics*, vol. 156, no. 17, pp. 3223–3237, Oct. 2008.

[9] E. Koutsoupias, C. Papadimitriou, and M. Yannakakis, "Searching a fixed graph," in *International Colloquium on Automata, Languages, and Programming*, Paderborn, Germany, 8–12 Jul. 1996, pp. 280–289.

[10] G. Ausiello, S. Leonardi, and A. Marchetti-Spaccamela, "On salesmen, repairmen, spiders, and other traveling agents," in *Proceedings of the 4th Italian Conference on Algorithms and Complexity*, 2000.

[11] A. Sarmiento, R. Murrieta-Cid, and S. Hutchinson, "A multi-robot strategy for rapidly searching a polygonal environment," in *9th Ibero-American Conference on AI*. Springer, 2004.

[12] T. Shermer, "Recent results in art galleries [geometry]," *Proceedings of the IEEE*, vol. 80, no. 9, pp. 1384 –1399, sep 1992.

[13] A. Sarmiento, R. Murrieta, and S. Hutchinson, "An efficient motion strategy to compute expected-time locally optimal continuous search paths in known environments," *Advanced Robotics*, 2009.

[14] M. Kulich, L. Přeučil, and J. J. M. Bront, "Single robot search for a stationary object in an unknown environment," in *International Conference on Robotics and Automation (ICRA)*, 2014.

[15] L. E. Wixson and D. H. Ballard, "Using intermediate objects to improve the efficiency of visual search," *International Journal of Computer Vision*, pp. 209–230, Apr. 1994.

[16] L. Kunze, K. K. Doreswamy, and N. Hawes, "Using qualitative spatial relations for indirect object search," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[17] M. Lorbach, S. Hfer, and O. Brock, "Prior-assisted propagation of spatial information for object search," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014.

[18] T. Krajník, J. Santos, B. Seemann, and T. Duckett, "FrOctomap: An efficient spatio-temporal environment representation," in *Proceedings of Towards Autonomous Robotic Systems (TAROS)*, 2014.

[19] T. Krajník *et al.*, "Long-term topological localization for service robots in dynamic environments using spectral maps," in *International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[20] D. M. Titterington, A. F. Smith, U. E. Makov *et al.*, *Statistical analysis of finite mixture distributions*. Wiley New York, 1985, vol. 7.

[21] D. B. Johnson, "Efficient algorithms for shortest paths in sparse networks," *J. ACM*, vol. 24, no. 1, pp. 1–13, Jan. 1977.

[22] D. J. Cook, "Learning setting-generalized activity models for smart spaces," *IEEE Intelligent Systems*, vol. 2010, no. 99, p. 1, 2010.

[23] T. Krajník *et al.*, "A practical multirobot localization system," *Journal of Intelligent and Robotic Systems*, 2014.

[24] R. Ambrus *et al.*, "Meta-rooms: Building and maintaining long term spatial models in a dynamic world," in *International Conference on Intelligent Robots and Systems (IROS)*, 2014.