

Grid-Based Models for Dynamic Environments

Daniel Meyer-Delius Maximilian Beinhofer Wolfram Burgard

Abstract—The majority of existing approaches to mobile robot mapping assume that the world is static, an assumption which does not hold in most practical application domains. In this paper we present a probabilistic grid-based approach for modeling dynamic environments representing both, the occupancy and the dynamics of the corresponding area. We describe the environment as a spatial grid and use a hidden Markov model to represent the occupancy state and state transition probabilities of each grid cell. Our approach updates the occupancy state as observations become available. We describe an offline and an online technique to estimate the transition probabilities of the model from observed data. Experimental results show that our model is better suited for representing dynamic environments than standard occupancy grids. Furthermore, the results show that the explicit representation of the environment dynamics can be used to improve robot navigation.

I. INTRODUCTION

An accurate model of the environment is essential for many mobile robot navigation tasks. Although the environment generally is dynamic, most existing navigation approaches assume it to be static. They typically build the map of the environment in an offline phase and then use it without considering potential future changes. There are robust approaches that can handle inconsistencies between the map and the actual measurements. However, a largely inconsistent model can lead to unreliable navigation or even to a complete localization failure.

In this paper we consider the problem of modeling a mobile robot’s environment taking the dynamics of the environment explicitly into account. We present a probabilistic model that represents the occupancy of the space and characterizes how this occupancy changes over time. The explicit representation of how the occupancy changes in time provides a better understanding of the environment that can be used to improve the navigation performance of the robot.

In our approach, we describe the environment as a spatial grid and use a hidden Markov model (HMM) to represent the belief about the occupancy state and state transition probabilities of each grid cell. Our model, called *dynamic occupancy grid*, is a generalization of a standard occupancy grid. Figure 1 illustrates the fundamental difference between these two models: while occupancy grids characterize the state of a cell as static, our representation explicitly models state changes.

All authors are with the University of Freiburg, Dept. of Computer Science, D-79110 Freiburg, Germany.

This work has been partially supported by the European Commission under contract numbers FP7-231888-EUROPA and FP7-260026-TAPAS. Also by the German Research Foundation (DFG) within the Research Training Group 1103.

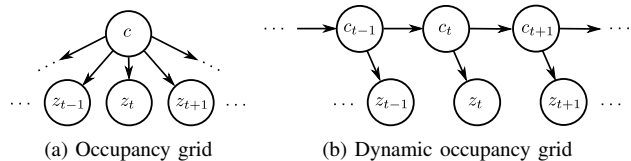


Fig. 1. Bayesian network describing the dependencies between the states of a cell c and observations z in standard and dynamic occupancy grids.

In addition to the explicit representation of the environment dynamics, the HMM framework provides efficient algorithms for estimating the model parameters. This allows us to learn the dynamics of the environment from observations made by the robot. Furthermore, within the framework we can efficiently estimate the occupancy state of a cell from the observed evidence as it becomes available, making it possible to adapt the representation continuously over time.

The contribution of this work is a mapping approach that represents the occupancy of the space and explicitly characterizes how this occupancy changes over time. We describe our model and how the representation can be updated as new observations become available. Furthermore, we present two techniques, one offline and one online, to estimate the state transition probabilities of the model from observed data. We evaluate our approach in simulation and using real-world data. The results demonstrate that our model can represent dynamic environments more accurately than standard occupancy grids. Furthermore, we show how the explicit representation of the environment dynamics can be used to improve the path planning performance of a robot.

II. RELATED WORK

Previous work on mapping dynamic environments can be divided into two groups: approaches that filter out sensor measurements caused by dynamic elements and approaches that explicitly model aspects of the environment dynamics. Filtering out sensor measurements is based on probabilistic sensor models that identify the measurements which are inconsistent with a reference model of the environment. Fox *et al.* [1], for example, use an entropy gain filter. Burgard *et al.* [2] propose a distance filter based on the expected distance of a measurement. Hähnel *et al.* [3] combine the EM algorithm and a sensor model that considers dynamic objects to obtain accurate maps. In contrast to these approaches, our work explicitly represents the dynamics of the environment in the environment’s model itself instead of relying on sensor models to represent them.

To model the dynamics of the environment, some authors have proposed augmented representations of the environment

which explicitly represent dynamic objects. The approaches of Anguelov *et al.* [4] and Biswas *et al.* [5], for example, compute shape models of non-stationary objects. They create maps at different points in time and compare those maps using an EM-based algorithm to identify the parts of the environment that change over time. Petrovskaya and Ng [6] extend occupancy grid maps with parameterized models of dynamic objects like doors and apply a Rao-Blackwellized particle filter to estimate the pose of the robot and the state of the dynamic objects. The above-mentioned approaches are based on the identification and modeling of dynamic objects in the environment. Our approach, in contrast, does not depend on high level object models and considers only the occupancy of the space at a lower level of abstraction. It is similar in spirit to the spatial affordance maps proposed by Luber *et al.* [7]. These maps represent space-dependent occurrences of relevant people activity events in the context of people tracking using Poisson processes. The fundamental difference to our model is that we consider the occupancy of the space and not people tracking events. For this reason our approach relies on HMMs that are better suited than Poisson processes to describe the behavior of the occupancy in a cell.

The problem of modeling the occupancy of the space in dynamic environments at a low level of abstraction has already been addressed in the past. Wolf and Sukhatme [8], for example, propose a model that maintains two separate occupancy grids, one for the static parts of the environment and the other for the dynamic parts. Brechtel *et al.* [9] describe a grid-based representation that in addition to the occupancy state of the cells, also stores a velocity vector that can be interpreted as the velocity of the object that occupies the cell. These approaches, however, are rather focused on the problem of tracking dynamic objects in the environment than on representing the environment’s dynamics. Biber and Duckett [10] propose a model that represents the environment on multiple timescales simultaneously. For each timescale a separate sample-based representation is maintained and updated using the observations of the robot according to an associated timescale parameter. Konolige and Bowman [11] describe a vision-based system where camera views are grouped into clusters that represent different persistent configurations of the environment. Changes in the environment are handled by deleting views based on a least-recently-used principle. The fundamental difference between previous approaches and ours is that, besides being able to continuously adapt to changes over time, our model provides an explicit characterization of the dynamics of the environment.

III. DYNAMIC OCCUPANCY GRIDS

Occupancy grids (as they were introduced by Moravec and Elfes [12]) are a regular tessellation of the space into a number of rectangular cells. They store in each cell the probability that the corresponding area of the environment is occupied by an obstacle. To avoid a combinatorial explosion of possible grid configurations, the approach assumes that neighboring cells are independent from each other.

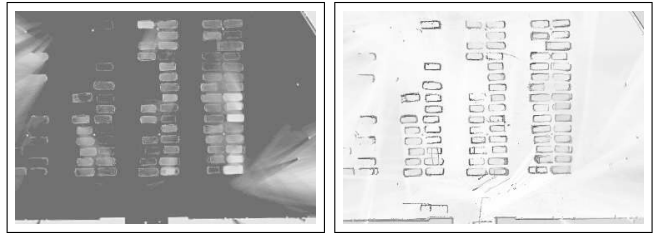


Fig. 2. State transition probabilities at the faculty’s parking lot. The left and right images correspond to the distributions $p(c_t = \text{free} | c_{t-1} = \text{free})$ and $p(c_t = \text{occ} | c_{t-1} = \text{occ})$ respectively. The darker the color, the larger the probability for the occupancy to remain unchanged.

Occupancy grids rest on the assumption that the environment is static. As mentioned above, they store for each cell c of an equally spaced grid, the probability $p(c)$ that c is occupied by an obstacle. Thus far, there is no model about how the occupancy changes over time. The approach described in this paper overcomes this limitation by relying on an HMM (see [13]) to explicitly represent both the belief about the occupancy state and state transition probabilities of each grid cell as illustrated in Figure 1.

An HMM requires the specification of a state transition, an observation, and an initial state distribution. Let c_t be a discrete random variable that represents the occupancy state of a cell c at time t . The initial state distribution or prior $p(c_{t=0})$ specifies the occupancy probability of a cell at the initial time step $t = 0$ prior to any observation.

The state transition model $p(c_t | c_{t-1})$ describes how the occupancy state of cell c changes between consecutive time steps. We assume that the changes in the environment are caused by a stationary process, that is, the state transition probabilities are the same for all time steps t . These probabilities are what allows us to explicitly characterize how the occupancy of the space changes over time. Since we are assuming that a cell c is either free (*free*) or occupied (*occ*), the state transition model can be specified using only two transition probabilities, namely $p(c_t = \text{free} | c_{t-1} = \text{free})$ and $p(c_t = \text{occ} | c_{t-1} = \text{occ})$. Note that, by assuming a stationary process, these probabilities do not depend on the absolute value of t . Figure 2 depicts transition probabilities for the parking lot at our faculty. The darker the color, the larger the probability for the corresponding occupancy to remain unchanged. The figure clearly shows the parking spaces, driving lanes, and static elements such as walls and lampposts as having different dynamics. The “shadows” in the upper left and lower right areas of the maps were mostly caused by maximum range measurements being ignored.

The observation model $p(z | c)$ represents the likelihood of the observation z given the state of the cell c . The observations correspond to measurements obtained with a range sensor. In this paper, we consider only observations obtained with a laser range scanner. The cells in the grid that are covered by a laser beam are determined using a ray-tracing operation. We consider two cases: the beam is not a maximum range measurement and ends up in a cell (a *hit*) or the beam covers a cell without ending in it (a *miss*). Accordingly, the observation model can also be

specified using only two probabilities: $p(z = \textit{hit} \mid c = \textit{free})$ and $p(z = \textit{hit} \mid c = \textit{occ})$. We additionally take into account the situation where a cell is not observed at a given time step. This is necessary since the transition model characterizes state changes only for consecutive time steps. Explicitly considering this *no-observation* case allows us to update and estimate the parameters of the model using the HMM framework directly without having to distinguish between observations and no-observations. The concrete observation probability for a *no-observation* does not affect the results as long as the proportion between the two remaining probabilities remains unchanged.

From the discussion above it can be seen that standard occupancy grids are a special case of dynamic occupancy grids where the transition probabilities $p(c_t = \textit{free} \mid c_{t-1} = \textit{free})$ and $p(c_t = \textit{occ} \mid c_{t-1} = \textit{occ})$ are 1 for all cells c .

A. Occupancy State Update

The update of the occupancy state of the cells in a dynamic occupancy grid follows a Bayesian approach. The goal is to estimate the belief or posterior distribution $p(c_t \mid z_{1:t})$ over the current occupancy state c_t of a cell given all the available evidence $z_{1:t}$ up to time t . The update formula is:

$$p(c_t \mid z_{1:t}) = \eta p(z_t \mid c_t) \sum_{c_{t-1}} p(c_t \mid c_{t-1}) p(c_{t-1} \mid z_{1:t-1}), \quad (1)$$

where η is a normalization constant. Exploiting the Markov assumptions in our HMM, this equation is obtained using Bayes' rule with $z_{1:t-1}$ as background knowledge and applying the theorem of total probability on $p(c_t \mid z_{1:t-1})$ conditioning on the state of the cell c_{t-1} at the previous time step $t-1$. Equation (1) describes a recursive approach to estimate the current state of a cell given a current observation and the previous state estimate. This approach corresponds to a discrete Bayes filter. The structure of our particular HMMs allows for a simple and efficient implementation of this approach. Note that the map update for standard occupancy grids is a special case, where the sum in (1) is replaced by the posterior $p(c_t \mid z_{1:t-1})$.

This posterior, corresponds to a prediction of the occupancy state of the cell at time t based on the observations up to time $t-1$. Prediction can be considered as filtering without the processing of evidence. By explicitly considering no-observations as explained in the previous section, the update formula can be used directly to estimate the future state of a cell or estimate the current state of a cell that has not been observed recently.

B. Parameter Estimation

As mentioned above, an HMM is characterized by the state transition probabilities, the observation model, and the initial state probabilities. We assume that the observation model only depends on the sensor. Therefore it can be specified beforehand and is the same for each HMM. We estimate the remaining parameters using observations that are assumed to correspond to the environment that is to be represented.

One of the most popular approaches for estimating the parameters of an HMM is an instance of the expectation-maximization (EM) algorithm. The basic idea is to iteratively estimate the model parameters using the observations and the parameters estimated in the previous iteration until the values converge. Let $\hat{\theta}^{(n)}$ represent the parameters estimated at the n -th iteration. The EM algorithm results in the following re-estimation formula for the transition model of cell c :

$$\hat{p}(c_t = i \mid c_{t-1} = j)^{(n+1)} = \frac{\sum_{\tau=1}^T p(c_{\tau-1} = i, c_{\tau} = j \mid z_{1:T}, \hat{\theta}^{(n)})}{\sum_{\tau=1}^T p(c_{\tau-1} = i \mid z_{1:T}, \hat{\theta}^{(n)}), \quad (2)$$

where $i, j \in \{\textit{free}, \textit{occ}\}$ and T is the length of the observation sequence used for estimating the parameters. Note that the probabilities on the right-hand side are conditioned on the observation sequence $z_{1:T}$ and the previous parameter estimates $\hat{\theta}^{(n)}$. These probabilities can be efficiently computed using the forward-backward procedure [13].

This, however, is an offline approach that requires storing the complete observation sequence for each cell. An online version of the algorithm was derived by Mongillo and Deneve [14]. To calculate the transition probabilities in (2), this algorithm only needs to store the sufficient statistics

$$\phi_{ijh}(t; \hat{\theta}) = \frac{1}{t} \sum_{\tau=1}^t \delta(z_{\tau}, h) p(c_{\tau-1} = i, c_{\tau} = j \mid z_{1:t}, \hat{\theta}), \quad (3)$$

where $i, j \in \{\textit{free}, \textit{occ}\}$, $h \in \{\textit{free}, \textit{occ}, \textit{no-observation}\}$, and $\delta(z_{\tau}, h) = 1$ if $z_{\tau} = h$ and 0 otherwise. Dropping the dependence on $\hat{\theta}$, only 16 values have to be stored. The algorithm uses ϕ_{ijh} instead of the probabilities computed with the forward-backward procedure to estimate the transition model. Therefore it implements only a partial expectation step, while the maximization step remains exact.

Besides being an online approach with small storage requirements, the prefactor $1/t$ in (3) allows the algorithm to handle non-stationary environment's dynamics. Additionally, the algorithm updates with each observation the occupancy state according to (1). These properties make the online version of the EM algorithm an attractive alternative for systems operating over extended periods of time.

IV. EXPERIMENTAL EVALUATION

We implemented our proposed model and tested it in simulation and using data obtained with a real robot. The goal of the experiments was to evaluate the quality and usefulness of the representation.

A. Accuracy of the Representation

In a first experiment we evaluated the accuracy of our proposed representation of the environment. We steered a MobileRobots Powerbot equipped with a SICK LMS laser range finder through the faculty's parking lot. We performed a run every full hour from 7am until 6pm during one day. The range data obtained from the twelve runs (data sets d_1 through d_{12}) corresponded to twelve different configurations of the parked cars, including an almost empty parking lot

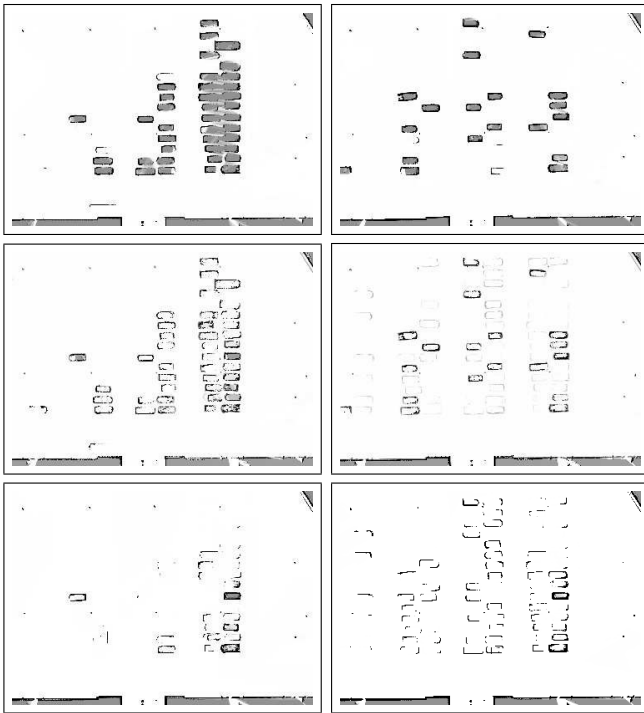


Fig. 3. Comparison between dynamic and standard occupancy grids. Shown are the ground truth (top), dynamic occupancy grid (middle), and standard occupancy grid (bottom) maps at two different points in time.

(data set d_1) and a relatively occupied one (data set d_{10}). We used a SLAM approach [15] to correct the odometry of the robot and obtain a good estimate of its pose. Range measurements were sampled at about 1 Hz, and the trajectory and velocity of the robot during each run were approximately the same to try to avoid a bias in the complete data set.

Figure 3 shows a qualitative comparison between dynamic and standard occupancy grids for the parking lot data set. The online EM approach was used to build the dynamic occupancy grid. We assumed that the parking lot did not change considerably during a run and used the occupancy grids obtained from every data set with the above-mentioned SLAM approach as ground truth. In the figure, the maps on the left column show the grids after the third run, that is, after integrating data sets d_1 through d_3 . The maps on the right show the grids at the end of the last run, after integrating data sets d_1 through d_{12} . As can be seen, the dynamic occupancy grid readily adapts to the changes in the parking lot. Thus it constitutes a better representation of the environment at any point in time. Additionally, dynamic occupancy grids provide information about the probable occupancy of areas that have not been recently observed. This appears in the grids in the figure (specially the right column) as light gray areas in the places where the cars most frequently park.

To quantitatively evaluate the accuracy of our representation we computed its accuracy with respect to the ground truth maps. In this context, accuracy is defined as the number of correctly classified cells divided by the number of all classified cells. A cell c was classified as *occupied* if $p(c) > 0.5$ and as *free* if $p(c) < 0.5$. The remaining cells were not taken into account. Figure 4 compares the accuracy of a

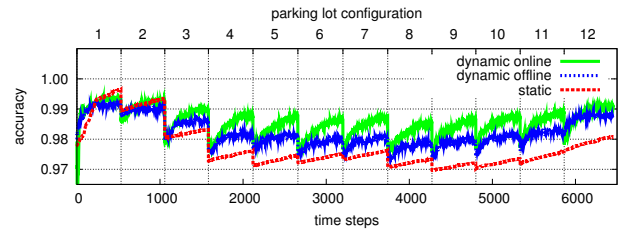


Fig. 4. Accuracy over time of standard and dynamic occupancy grids for the parking lot data. Both online and offline parameter estimation approaches were evaluated.

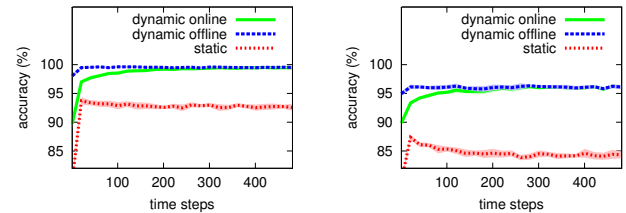


Fig. 5. Accuracy of standard and dynamic occupancy grids for different configurations of dynamic cells in a simulated grid map. Left: 5% dynamic cells and 5% state change probability. Right: 25% dynamic cells and 25% state change probability

standard occupancy grid (*static*) against that of the dynamic occupancy grid whose parameters were estimated online (*dynamic online*). We additionally consider the case when the parameters were estimated offline (*dynamic offline*). Figure 2 depicts the obtained parameters. Figure 4 plots the accuracy of the grids over time for the parking lot data. After each configuration change, the accuracy of the dynamic occupancy grids quickly starts to increase as the map adapts to the new configuration. Standard occupancy grids adapt relatively quickly at first, but their adaptability decreases with the number of observations already integrated into the map.

B. Effects of the Environment's Dynamics

The accuracy of dynamic occupancy grids and their advantage over standard occupancy grids strongly depends on the environment's dynamics. In the parking lot environment of the previous experiment, only a small number of cells were dynamic ($\sim 3\%$) and only few changes took place. This explains why the standard occupancy grids in Figure 3 corresponding respectively to configurations 3 and 12 in Figure 4 have high accuracy values even though they are evidently inaccurate.

The goal of this experiment was to evaluate the accuracy of our proposed representation for different environment dynamics. In the context of occupancy grids, the dynamics of the environment are characterized by the number of dynamic cells and their state change probabilities. We used a 50×50 grid map and changed the fraction of dynamic cells and state change probabilities to generate artificial data for different environment dynamics. We estimated the state transition probabilities of the dynamic occupancy grid using both the online and offline approaches and compared the resulting model accuracy against that of a standard occupancy grid for different data sets. Figure 5 shows the results for two

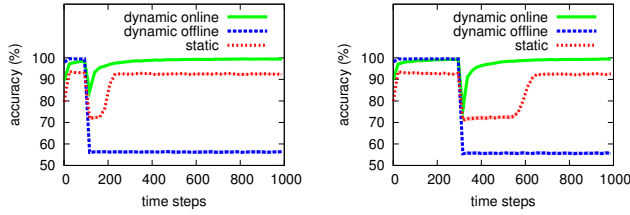


Fig. 6. Accuracy of the parameters obtained with the online and offline approaches when the environment’s dynamics change. The left and right plots correspond to changes after 100 and 300 time steps respectively.

different settings: one relatively static and another more dynamic. The curves correspond to the mean and standard deviation for 10 repetitions of the experiment. As can be seen in the figure, dynamic occupancy grids represent the environment more accurately than standard occupancy grids. Nevertheless, the more static the environment, the smaller the difference between the accuracies. This experiment shows that even for moderately dynamic environments dynamic occupancy grids outperform standard occupancy grids.

C. Parameter Estimation

As can be seen in Figure 5, the offline approach produces more accurate results at first, but the difference between the results of the two approaches decreases over time. This suggests that, regarding the accuracy of the representation over time, both parameter estimation techniques are comparable.

Although the offline approach produces good results from the beginning, it requires storing all observations for each cell in the grid for an a priori training phase. This is a considerable disadvantage since it limits the amount of data that can be used for training, the resolution of the grid, or the size of the environment that can be represented. Furthermore, being an offline approach, once the parameters have been estimated, they remain fixed. This makes the offline approach inappropriate for environments where the assumption that the environment dynamics are stationary does not hold. In contrast, the online approach continually adapts its parameters as new observations become available. Figure 6 illustrates the effects on the accuracy of the model parameters obtained with the two approaches for 5% dynamic cells and 5% state change probability (left plot in Figure 5) in the case that the environment dynamics change. For the experiment, the change consisted in selecting a new set of dynamic and static cells. The number of dynamic cells and their state change probabilities remained the same, but we obtained similar results when these parameters were changed as well. As can be seen in the figure, using the online approach, the accuracy of the model quickly returns to its value before the change. This is the result of the model parameters adapting to the new environment dynamics. The accuracy of the model whose parameters were estimated offline drops when the dynamics change, and remains low. We also evaluated the behavior of standard occupancy grids. As expected, the number of observations needed by a standard occupancy grid to correctly represent the occupancy of the new static cells is approximately the same as the number of previous



Fig. 7. Experimental setup for the path planning experiment. The task consists in navigating between *A* and *B*. At *C* we added a virtual door that changed its state over time.

observations. Note that the dynamic cells remain inaccurately represented.

D. Path Planning Using Dynamic Occupancy Grids

In the previous experiments we showed that dynamic occupancy grids readily adapt to changes in the environment. The goal of this experiment was to show that this adaptability can be used to improve the path planning performance of a robot. The experiment was performed in simulation within the environment shown in Figure 7 corresponding to part of the Intel Research Lab in Seattle. The task of the robot was to navigate between positions *A* and *B*. We added a virtual door at position *C* along the shortest path between *A* and *B*. The state of the door changed each time step with a probability of 0.001. To generate the a priori map needed for path planning and obtain data for estimating the parameters of the dynamic occupancy grid, we steered the robot through the relevant parts of the environment in an offline phase.

We then performed 20 repetitions of the experiment. In every repetition, the robot executed 20 runs from one position to the other. The A^* algorithm was used for path planning and re-planning was performed at every time step. The cost of a path was computed as the sum of the traversal costs for each cell in the path. The traversal cost was set to 1 for *free* cells and infinity for *occupied* cells. The cells in the grid were classified as described in the first experiment.

Since numeric performance measures, like traveled distance or execution time, largely depend on the particular environment used for the experiment, we opted for a more qualitative evaluation and classified the trajectories followed by the robot into five types:

- 1) *short*: the robot followed the shortest path.
- 2) *long*: the robot followed the longer path. In this case, following the longer path was the optimal choice.
- 3) *indirect long*: the robot tried to follow the shortest path first, found the door closed, turned around, and finally followed the longer path. In this case, following the longer path from the beginning would have been the optimal choice.
- 4) *unnecessary indirect long*: as in the previous case, the robot tried to follow the shortest path first, found the door closed, turned around, and finally followed the longer path. In this case, however, continuing to follow the shortest path would have been the optimal choice since the door would have opened for the robot to pass.
- 5) *unnecessary direct long*: the robot followed the longer path. No attempt was made to follow the shortest path. In this case, following the shortest path from the beginning would have been the optimal choice.

TABLE I
OCCURRENCES OF TRAJECTORY TYPES FOR DIFFERENT OCCUPANCY GRIDS AND PARAMETER ESTIMATION APPROACHES.

values in %	<i>static</i>	<i>dynamic online</i>	<i>dynamic offline</i>	<i>random</i>	<i>optimal</i>
<i>short</i>	11.25 (± 16.35)	25.50 (± 22.24)	40.75 (± 20.15)	24.25 (± 10.17)	40.50 (± 19.99)
<i>long</i>	35.75 (± 14.80)	27.00 (± 18.38)	15.00 (± 9.03)	24.25 (± 13.70)	17.25 (± 9.39)
<i>indirect long</i>	4.75 (± 1.12)	19.50 (± 12.24)	30.50 (± 8.72)	23.50 (± 9.33)	27.75 (± 9.24)
<i>unnecessary indirect long</i>	0.25 (± 1.12)	1.50 (± 2.86)	2.75 (± 3.02)	1.50 (± 2.86)	2.25 (± 3.02)
<i>unnecessary direct long</i>	48.00 (± 16.89)	26.50 (± 21.34)	11.00 (± 5.28)	26.50 (± 13.19)	12.25 (± 6.38)
accuracy	47.00 (± 16.89)	52.50 (± 16.10)	55.75 (± 13.21)	48.50 (± 10.14)	57.75 (± 14.09)

The values in Table I correspond to the occurrences (average and standard deviation) of the different trajectory types during the 20 repetitions of the experiment. We compared the path planning performance when using a standard occupancy grid (*static*), a dynamic occupancy grid whose parameters were estimated online (*dynamic online*), and a dynamic occupancy grid whose parameters were estimated offline (*dynamic offline*). The number of occurrences of *short* and *long* trajectories in the table indicate that the information about the state change probability of the door, represented in the dynamic occupancy grid, leads to better path planning performances. Once the door is represented as closed in a standard occupancy grid, the robot never attempts to follow the shortest path again. This, in turn, prevents the robot from updating the cells corresponding to the door. This can be seen in the table by the small number of *short*, *indirect long*, and *unnecessary indirect long* trajectories and explains the large number of *long* trajectories. Using dynamic occupancy grids, on the other hand, the state of the (unobserved) door in the map changes over time according to the learned state transition probabilities. Whenever the cells corresponding to the door are classified as free, the robot attempts to follow the shortest path.

We additionally implemented two baseline path planning policies for comparison. In the first (*random*) the robot, when in *A* or *B*, randomly chooses between the two possible paths and never replans while on the way. In the second (*optimal*) the robot has perfect knowledge about the state change probability of the door and based on its internal belief about the state of the door chooses the expected optimal path. The percentage of runs in which the optimal path was followed (*accuracy*) by this robot is an empirical upper bound for the accuracy achievable in our experimental setup. One sided t-tests show that the accuracy for *dynamic offline* was significantly higher than those for *static* and *random* on a 5% level. Between the accuracies for *dynamic offline* and *optimal* we could not find a significant difference.

V. CONCLUSIONS

In this paper we introduced a novel approach to occupancy grid mapping that explicitly represents how the occupancy of individual cells changes over time. Our model is a generalization of standard occupancy grids. It applies HMMs to update the belief about the occupancy state of each cell according to the dynamics of the environment. We described how our maps can be updated as new observations become available. We furthermore introduced an offline and an online technique to estimate the parameters of the model from

observed data. We evaluated our approach in simulation and using real-world data. The results demonstrate that our model can represent dynamic environments more accurately than standard occupancy grids. We also demonstrated that using our model can improve the path planning performance of a robot. The representation and online parameter estimation approach presented in this paper can, in principle, also be combined with a Rao-Blackwellized particle filter [15] to perform SLAM, thus enabling long-term operation of mobile robots in dynamic environments.

REFERENCES

- [1] D. Fox, W. Burgard, and S. Thrun, "Markov localization for mobile robots in dynamic environments," *Journal of Artificial Intelligence Research*, vol. 11, 1999.
- [2] W. Burgard, A. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "Experiences with an interactive museum tour-guide robot," *Artificial Intelligence*, vol. 114, no. 1-2, 2000.
- [3] D. Hähnel, R. Triebel, W. Burgard, and S. Thrun, "Map building with mobile robots in dynamic environments," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2003.
- [4] D. Anguelov, R. Biswas, D. Koller, B. Limketkai, S. Sanner, and S. Thrun, "Learning hierarchical object maps of non-stationary environments with mobile robots," in *Proc. of the Conference on Uncertainty in AI (UAI)*, 2002.
- [5] R. Biswas, B. Limketkai, S. Sanner, and S. Thrun, "Towards object mapping in non-stationary environments with mobile robots," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2002.
- [6] A. Petrovskaya and A. Y. Ng, "Probabilistic mobile manipulation in dynamic environments, with application to opening doors," in *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, 2007.
- [7] M. Luber, G. D. Tipaldi, and K. O. Arras, "Place-dependent people tracking," in *Proc. of the Int. Symposium of Robotics Research (ISRR)*, 2009.
- [8] D. F. Wolf and G. S. Sukhatme, "Mobile robot simultaneous localization and mapping in dynamic environments," *Autonomous Robots*, vol. 19, no. 1, pp. 53–65, 2005.
- [9] S. Brechtel, T. Gindele, and R. Dillmann, "Recursive importance sampling for efficient grid-based occupancy filtering in dynamic environments," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2010.
- [10] P. Biber and T. Duckett, "Dynamic maps for long-term operation of mobile service robots," in *Proc. of Robotics: Science and Systems (RSS)*, 2005.
- [11] K. Konolige and J. Bowman, "Towards lifelong visual maps," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009.
- [12] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 1985.
- [13] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in *Proceedings of the IEEE*, vol. 77 (2), 1989, pp. 257–286.
- [14] G. Mongillo and S. Deneve, "Online learning with hidden Markov models," *Neural Computation*, vol. 20, pp. 1706–1716, 2008.
- [15] G. Grisetti, C. Stachniss, and W. Burgard, "Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2005.