Simultaneous Localization and Dynamic State Estimation in Reconfigurable Environments

Gian Diego Tipaldi Daniel Meyer-Delius Maximilian Beinhofer Wolfram Burgard

***Abstract*—The majority of existing approaches to mobile robot localization assume that the world is static, which clearly does not hold in most real-world application domains. In this paper we present a probabilistic approach to global localization in reconfigurable environments, where the robot pose and the environment state are jointly estimated using a Rao-Blackwellized particle filter. The environment is represented as a spatial grid and a hidden Markov model is used to represent the occupancy state and state transition probabilities of each grid cell. The HMM parameters are estimated offline using the EM algorithm. Experimental results show that our model is better suited for representing reconfigurable environments than standard occupancy grids. Furthermore, the results show that the explicit representation of the environment dynamics can be used to improve localization accuracy in reconfigurable environments.**

I. Introduction

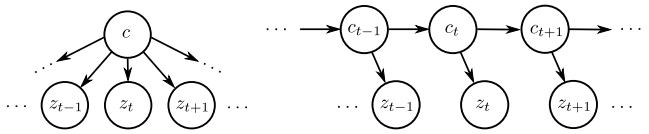
An accurate model of the environment is essential for many mobile robot navigation tasks. Although the environment generally is dynamic, most existing navigation approaches assume it to be static. They typically build the map of the environment in an offline phase and then use it without considering 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. Moreover, some areas can be *semi-static* (e.g., furnitures can move, cars can be parked) and this information can be used by the robot to improve its navigation performance.

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. We then show how this information can be used to jointly estimate the pose of the robot and the configuration of the environment during a global localization task.

The environment is represented as a two-dimensional grid where each cell uses a hidden Markov model (HMM) to represent the belief about the occupancy state and state transition probabilities. Our model, called *dynamic occupancy grid*, is a generalization of a standard occupancy grid. Figure 1 illustrates the fundamental difference between these two models:

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.



(a) Occupancy grid (b) Dynamic occupancy grid

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

while occupancy grids characterize the state of a cell as static, our representation explicitly models state changes.

In addition to the explicit representation of the environment dynamics, the HMM framework provides efficient algorithms for estimating the model parameters. Given that, we can use a Rao-Blackwellized particle filter (RBPF) to jointly reason about the robot pose, which represents the sampled part of the filter, and the occupancy probability of a cell, represented in the analytical part of the factorization.

The contribution of this work is a global localization approach that estimates the pose of the robot and, at the same time, explicitly infers how the state of the world changes over time. To the best of our knowledge, this is the first approach to address this problem. Previous attempts either focused on how to filter spurious observations due to dynamic objects or addressed the easier problem of pose tracking. We describe our model and how the representation can be updated as new observations become available. We further propose a local map representation that is able to forget changes in a sound probabilistic way, using the mixing times of the associated Markov chain, and to minimize memory requirements.

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 that it outperforms standard Monte Carlo localization in complex real world environments with consistent changes over time.

II. Related Work

Most mobile robot navigation systems rely on a map of the environment built beforehand in an offline phase. To deal with subsequent changes in the environment, sensor measurements caused by dynamic objects are usually filtered out. Robust approaches rely on probabilistic sensor models that identify the measurements inconsistent with the map. For example, Fox *et al.* [1] use an entropy gain filter, while Burgard *et al.* [2] use a distance filter based on the expected distance of a measurement. Despite the success of these techniques, they

discard valuable information about the environment. Instead of filtering out inconsistent measurements, Montemerlo *et al.* [3], use them for people tracking while localizing the robot. They show that the state of dynamic objects in the environment can be used to increase the robustness of the pose estimation process. Motivated by this idea, we utilize all sensor measurements to keep the map of the environment up-to-date while simultaneously localizing the robot within the updated map.

Orthogonal to the work on localization in dynamic environments, many authors have addressed the problem of modeling such environments. Hähnel *et al.* [4], for example, combine the EM algorithm and a sensor model that considers dynamic objects to obtain accurate maps. The approach of Anguelov *et al.* [5] computes 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. Wolf and Sukhatme [6] propose a model that maintains two separate occupancy grids, one for the static parts of the environment and the other for the dynamic parts. Biber and Duckett [7] 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. In our work, we use the grid-based representation described by Meyer-Delius *et al.* [8] to represent dynamic environments. Besides being able to continuously adapt to changes over time, this model provides also an explicit characterization of the dynamics of the environment that can be learned from data.

Whereas most of the work on mapping dynamic environments assumes that a good estimate of the robot's pose is available, most of the work on mapping where the pose of the robot is not available (i.e., SLAM) assumes that the environment is static. Only few authors address the problem of jointly estimating the pose of the robot and the state of a dynamic environment. Avots *et al.* [9], for example, use a Rao-Blackwellized particle filter to estimate the pose of the robot and the state of doors in the environment. They represent the environment using a reference occupancy grid where the location of the doors is known, but not their state (i.e., opened of closed). Petrovskaya and Ng [10] propose a similar approach where instead of a binary model, a parametrized model (i.e., opening angle) of the doors is used. Similar to these approaches, we also use a Rao-Blackwellized particle filter to estimate the pose of the robot and the state of the environment. In contrast to their methods, however, we estimate the state of the complete environment, and not only of small, specific areas or elements. Additionally, we also learn the model parameters from data.

III. Dynamic Occupancy Grid

Occupancy grids (as they were introduced by Moravec and Elfes [11]) 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.

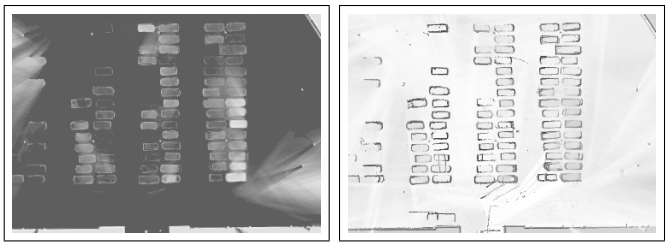


Fig. 2. State transition probabilities of the parking lot of the University of Freiburg. The left and right images correspond to the distributions $p(c_t = f \mid c_{t-1} = f)$ and $p(c_t = o \mid c_{t-1} = o)$ respectively. The darker the color, the larger the probability for the occupancy to remain 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. To probabilistically model how the occupancy changes over time in dynamic environments, the approach relies on an HMM (see Rabiner [12]) 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 $p(c_t \mid c_{t-1})$ 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 (*f*) or occupied (*o*), the state transition model can be specified using only two transition probabilities, namely $p(c_t = f \mid c_{t-1} = f)$ and $p(c_t = o \mid c_{t-1} = o)$. 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 $p(z \mid c)$ 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: $p(z = \textit{hit} \mid c = f)$ and $p(z = \textit{hit} \mid c = o)$. 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 $p(c_t = f \mid c_{t-1} = f)$ and $p(c_t = o \mid c_{t-1} = o)$ 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:

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

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 $i, j \in \{f, o\}$ 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 [12].

For more details about this model we refer the reader to the associated technical report by Meyer-Delius *et al.* [8].

IV. Simultaneous Localization and Dynamic State Estimation

In this section we describe our approach to simultaneously estimate the robot pose and the dynamic state of the environment. Although on first sight one can see the addressed problem as an instance of the better known simultaneous localization and mapping (SLAM), there are two main differences between them.

The first difference is the absence of a global reference frame in the SLAM problem. No global pose is required and the initial pose of the robot can typically be set freely. On the contrary, we explicitly address global localization as part of the estimation aspect. We have a global reference frame and the initial pose of the robot is unknown and assumed to be uniformly distributed over the whole environment. The second difference regards the dimensionality of the map. In the SLAM problem, the size of the map is not known in advance and can grow unbounded with time. In our problem the size of the map is known and we only focus on estimating the actual configuration of the dynamic objects present in the environments. Despite the differences, the two problems do share the same state space, i.e., the robot pose and the state of the map, and one can exploit the same factorization that made Rao-Blackwellized particle filters a feasible solution to the SLAM problem [13], [14].

In the following we show how this factorization can be exploited and we derive the RBPF that will be used to estimate the posterior $p(x_{1:t}, m_{1:t} \mid z_{1:t}, u_{1:t-1}, m_0)$ about the trajectory $x_{1:t}$ of the robot and the configuration of the environment $m_{1:t}$, given the observations $z_{1:t}$, the odometry measurements $u_{1:t-1}$ and the prior over the map m_0. The key

idea is to separate the estimation of the robot pose from the map estimation process,

$$p(x_{1:t},m_{1:t} \mid z_{1:t}, u_{1:t-1}, m_0) = p(m_{1:t} \mid x_{1:t}, z_{1:t}, m_0)p(x_{1:t} \mid z_{1:t}, u_{1:t-1}, m_0). \quad (3)$$

This can be done efficiently, since the posterior over maps $p(m_{1:t} \mid x_{1:t}, z_{1:t}, m_0)$ can be computed analytically given the knowledge of $x_{1:t}$ and $z_{1:t}$ and using the forward algorithm for the HMM. The remaining posterior, $p(x_{1:t} \mid z_{1:t}, u_{1:t-1})$, is estimated using a particle filter which incrementally processes the observations and the odometry readings. Following [15], we obtain a sample of the robot trajectory by incrementally sampling the current pose from the motion model $x_t^{(i)} \sim p(x_t|x_{t-1}^{(i)}, u_{t-1})$ and setting $x_{1:t}^{(i)} = \{x_t^{(i)}, x_{1:t-1}^{(i)}\}$. This recursive sampling schema allows us to recursively compute the importance weights using the following equation

$$\begin{aligned} w_t^{(i)} &= w_{t-1}^{(i)} \frac{p(z_t \mid x_{1:t}^{(i)}, z_{1:t-1}, m_0)p(x_t^{(i)} \mid x_{t-1}^{(i)}, u_{t-1})}{p(x_t^{(i)} \mid x_{t-1}^{(i)}, u_{t-1})} \\ &= w_{t-1}^{(i)} p(z_t \mid x_{1:t}^{(i)}, z_{1:t-1}, m_0). \end{aligned} \quad (4)$$

The observation likelihood is then computed by marginalization over the predicted state of the map leading to

$$\begin{aligned} p(z_t \mid x_{1:t}^{(i)}, z_{1:t-1}, m_0) &= \\ &= \int p(z_t \mid x_t^{(i)}, m_t) p(m_t \mid m_{t-1}^{(i)}) dm_t \\ &= \prod_j \mathcal{N}(z_t^j; \hat{z}_t^j, \sigma^2), \end{aligned} \quad (5)$$

were z^j is an individual laser reading and $\hat{z}_t^j$ is the closest cell in the map to the reading, with an occupancy probability above a certain threshold. Note that the *disappearance* of the integral is not an approximation but a direct consequence of using the likelihood field model described in [16].

A. Map Management

As we already mentioned above, the initial pose of the robot is unknown and assumed to be uniformly distributed in the whole environment. This forces us to use a high number of particles, generally above thousands, to accurately represent the initial distribution. Since every particle needs to have its own estimate of the map, memory management is a key aspect of the whole algorithm. In order to save memory, we want to only store the cells in the map that have been considerably changed from the a priori map m_0, which is shared among the diverse particles. This is done exploiting two important aspects of the Markov chain associated to the HMM: the *stationary distribution* and the *mixing time*.

As the number of time steps for which no observation is available tends to infinity, the occupancy value of a cell converges to a unique stationary distribution π (see [17]). This stationary distribution represents the case where the environment has not been observed for a long time and is represented by our a priori map m_0. In the case of a binary HMM, the one used in this paper, this distribution is computed using the transition probabilities

$$\begin{bmatrix} \pi_f \\ \pi_o \end{bmatrix} = \frac{1}{p+q} \begin{bmatrix} q \\ p \end{bmatrix}, \quad (6)$$

where for notation simplicity we have

$$\begin{aligned} p &= p(c_t = o \mid c_{t-1} = f) \\ q &= p(c_t = f \mid c_{t-1} = o). \end{aligned}$$

Every time an individual particle observes the state of a cell for the first time, the state distribution of that particular cell changes from the stationary one and the particle needs to store the new state of the cell. In order to reduce memory requirements, only a limited number of cells should be stored and a *forgetting* mechanism should be implemented. This can be done in a sound probabilistic way, by exploiting the mixing time of the associated Markov chain. The mixing time is defined as the time needed to converge from a particular state to the stationary distribution. The concrete definition depends on the measure used to compute the difference between distributions. In this paper we use the total variation distance as defined by Levin *et al.* [17]. Since our HMMs have only two states, the total variation distance Δ_t between the stationary distribution π and the occupancy distribution p_t at time t can be specified as

$$\Delta_t = |1 - p - q|^t \Delta_0, \quad (7)$$

where $\Delta_0 = |p(c_t = f) - \pi_f| = |p(c_t = o) - \pi_o|$ is the difference between the current state $p(c_t)$ and stationary distribution π. Based on the total variation distance, we can define the mixing time t_m as the smallest t such that the distance Δ_{t_m} is less than a given value ϵ. This leads to

$$t_m = \left\lceil \frac{ln(\epsilon/\Delta_0)}{ln(|1 - p - q|)} \right\rceil. \quad (8)$$

In other words, the mixing time tells us how many steps are needed for a particular cell to return to its stationary distribution, that is how many step a particle needs to store an unobserved cell before removing it from its local state and rely on the a priori map m_0.

V. Experiments

We implemented our proposed model and tested it using data obtained with a real robot. We steered a MobileRobots Powerbot equipped with a SICK LMS laser range finder through the parking lot of our faculty. 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}) corresponds to twelve different configurations of the parked cars, including an almost empty parking lot (data set d_1) and a relatively occupied one (data set d_8). We used a SLAM approach [18] 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.

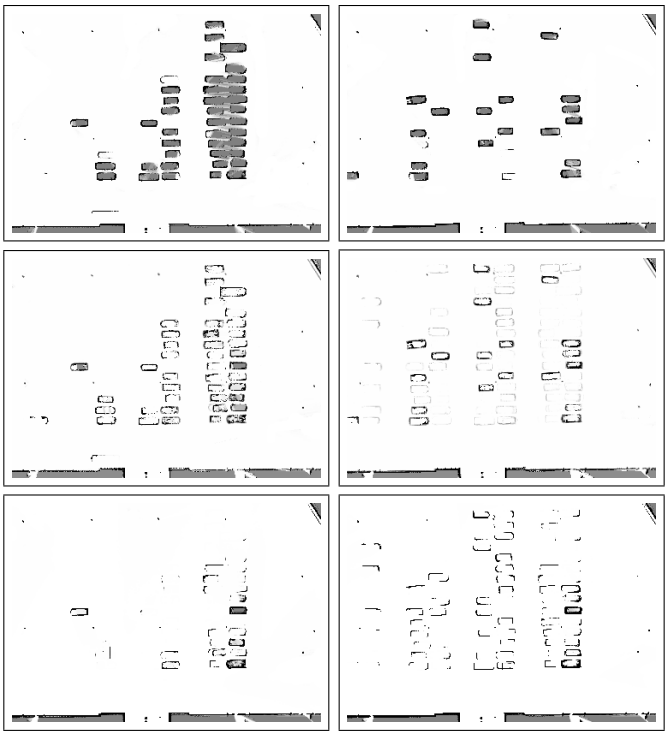


Fig. 3. Comparison between dynamic and standard occupancy grids. Shown are the ground truth (top), dynamic occupancy grid (middle), and standard occupancy grid (bottom) maps at two different points in time.

Figure 3 shows a qualitative comparison between dynamic and standard occupancy grids for the parking lot data set. 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 occupancy probability 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.

In order to assess the performances of the localization approach, we compared it to standard Monte Carlo localization both in a global localization and pose tracking setting. For each data set, we compared our approach (MCL-D), MCL using the standard occupancy grid (MCL-S), and MCL using the ground-truth map for that specific data set (MCL-GT). We performed 100 runs for each data set, where we randomly sampled the initial pose of the robot. In order to obtain a fair comparison, the same seed has been used to generate the initial pose, as well as to perform all the random sampling processes for each approach. All the approaches have been initialized with $10,000$ particles for global localization and 500 particles for pose tracking. They also used the same set of parameters, an occupancy threshold of 0.6 and a maximum distance of 1m for the likelihood field model.

The results of the global localization experiment are shown in Table I. The table shows the success rate of the global localization, as the percentage of time the filter converged to the true pose, and the residual squared error, with respective variance, after convergence. The success rate is reported relative to the result of MCL on the ground-truth map, in order to have a measure independent of the complexity of the environment. The results show that our approach outperforms the standard MCL on static maps both in terms of convergence rate and accuracy in localization.

Table II shows the results for the pose tracking experiment, where the filter is initialized at the true pose and keeps tracking the robot. The table shows the failure rate, i.e., the percentage of time the robot got lost during tracking, as well as the residual squared error. The results of this experiment show that the performances of the dynamic maps in pose tracking are almost equivalent to MCL with the ground-truth maps, with a failure rate of only 2%.

Both experiments show two important aspects of the problem and of the solution adopted. The first aspect is that the problem is much more complex than global localization since the search space is bigger and deciding if a measurement is an outlier or is caused by a change of the configuration is not a trivial task. Furthermore, analyzing the performances in pose tracking, we see that if the filter is initialized close to the correct solution, i.e., the search is reduced to the correct subspace, it is able to estimate the correct configuration. The second aspect is how the algorithm scales with different amount of change in the environment configuration. In the first four data sets, the parking lot is almost empty and it becomes quite full in the last ones. This is evident, when analyzing the results of MCL on the static maps, since the performance gets worse with an increasing amount of change. On the other hand, the performance of MCL on the dynamic maps is independent from the amount of change, as can be seen from the tables.

VI. Conclusions

In this paper we introduced a novel approach to global localization in reconfigurable environments using a grid map that explicitly represents how the occupancy of individual cells changes over time. The model is a generalization of standard occupancy grids and applies HMMs to update the belief about the occupancy state of each cell according to the dynamics of the environment. We described how a Rao-Blackwellized particle filter can be used to jointly estimate the robot pose and the configuration of the environment. This was possible thanks to the reduced memory requirements obtained by exploiting the properties of the Markov chains. We evaluated our approach using real-world data. The results demonstrate that our model can represent dynamic environments more accurately than standard occupancy grids and outperforms standard MCL on static maps in both global localization and pose tracking.

Data set	MCL-GT			MCL-D			MCL-S		
	Success	Error^2	σ^2	Success	Error^2	σ^2	Success	Error^2	σ^2
01	100%	0.21	0.36	50%	0.26	0.36	50%	0.26	0.18
02	100%	0.19	0.29	40%	0.10	0.08	33%	0.13	0.09
03	100%	0.13	0.19	80%	0.10	0.29	52%	0.19	0.17
04	100%	0.04	0.03	60%	0.08	0.14	53%	0.15	0.19
05	100%	0.07	0.18	54%	0.07	0.09	35%	0.15	0.18
06	100%	0.02	0.01	87%	0.02	0.02	45%	0.06	0.02
07	100%	0.06	0.08	59%	0.12	0.22	43%	0.14	0.20
08	100%	0.05	0.10	71%	0.03	0.02	28%	0.03	0.01
09	100%	0.02	0.01	53%	0.12	0.22	31%	0.06	0.02
10	100%	0.14	0.28	62%	0.13	0.31	34%	0.30	1.01
11	100%	0.11	0.11	38%	0.15	0.21	26%	0.24	0.29
12	100%	0.19	0.32	20%	0.16	0.14	22%	0.27	0.38
Total	100%	0.11	0.19	52%	0.11	0.18	36%	0.17	0.22

TABLE I
GLOBAL LOCALIZATION EXPERIMENT

Data set	MCL-GT			MCL-D			MCL-S		
	Failure	Error^2	σ^2	Failure	Error^2	σ^2	Failure	Error^2	σ^2
01	0%	0.04	0.01	3%	0.09	0.03	5%	0.18	0.07
02	0%	0.03	0.01	4%	0.08	0.05	24%	0.18	0.10
03	0%	0.04	0.01	2%	0.05	0.04	10%	0.09	0.04
04	0%	0.02	0.01	0%	0.04	0.01	10%	0.08	0.02
05	0%	0.02	0.01	3%	0.03	0.04	13%	0.06	0.02
06	0%	0.02	0.01	2%	0.02	0.01	26%	0.09	0.12
07	0%	0.02	0.01	0%	0.03	0.01	34%	0.07	0.01
08	0%	0.02	0.01	2%	0.02	0.01	35%	0.09	0.15
09	0%	0.02	0.01	4%	0.03	0.01	37%	0.07	0.16
10	0%	0.02	0.01	0%	0.03	0.01	36%	0.09	0.10
11	0%	0.03	0.01	1%	0.05	0.02	42%	0.10	0.05
12	0%	0.03	0.01	5%	0.06	0.01	44%	0.15	0.20
Total	0%	0.03	0.01	2%	0.04	0.02	27%	0.10	0.08

TABLE II
POSITION TRACKING EXPERIMENT

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] M. Montemerlo, S. Thrun, and W. Whittaker, "Conditional particle filters for simultaneous mobile robot localization and people-tracking," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2002.

[4] 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.

[5] 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.

[6] 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.

[7] P. Biber and T. Duckett, "Dynamic maps for long-term operation of mobile service robots," in *Proc. of Robotics: Science and Systems (RSS)*, 2005.

[8] D. Meyer-Delius, M. Beinhofer, and W. Burgard, "Grid-based models for dynamic environments," Dept. of Computer Science, University of Freiburg, Tech. Rep., 2011.

[9] D. Avots, E. Lim, R. Thibaux, and S. Thrun, "A probabilistic technique for simultaneous localization and door state estimation with mobile robots in dynamic environments," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2002.

[10] 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.

[11] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 1985.

[12] 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.

[13] K. Murphy, "Bayesian map learning in dynamic environments," in *Proc. of the Conf. on Neural Information Processing Systems (NIPS)*, Denver, CO, USA, 1999, pp. 1015–1021.

[14] G. Grisetti, C. Stachniss, and W. Burgard, "Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.

[15] A. Doucet, J. de Freitas, K. Murphy, and S. Russel, "Rao-Blackwellized partcile filtering for dynamic bayesian networks," in *Proc. of the Conf. on Uncertainty in Artificial Intelligence (UAI)*, Stanford, CA, USA, 2000, pp. 176–183.

[16] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[17] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov Chains and Mixing Times*. American Mathematical Society, 2008.

[18] 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.