<http://www.diva-portal.org>

Postprint

This is the accepted version of a paper presented at *IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, Vilamoura, Portugal, 7-12 October, 2012.

Citation for the original published paper:

Saarinen, J., Andreasson, H., Lilienthal, A. (2012)

Independent Markov Chain Occupancy Grid Maps for Representation of Dynamic Environments

In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 3489-3495). New York, USA: IEEE

IEEE International Conference on Intelligent Robots and Systems

<https://doi.org/10.1109/IROS.2012.6385629>

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:oru:diva-65747>

Independent Markov Chain Occupancy Grid Maps for Representation of Dynamic Environments

Jari Saarinen¹, Henrik Andreasson², and Achim J. Lilienthal²

Abstract—In this paper we propose a new grid based approach to model a dynamic environment. Each grid cell is assumed to be an independent Markov chain (iMac) with two states. The state transition parameters are learned online and modeled as two Poisson processes. As a result, our representation not only encodes the expected occupancy of the cell, but also models the expected dynamics within the cell. The paper also presents a strategy based on recency weighting to learn the model parameters from observations that is able to deal with non-stationary cell dynamics. Moreover, an interpretation of the model parameters with discussion about the convergence rates of the cells is presented. The proposed model is experimentally validated using offline data recorded with a Laser Guided Vehicle (LGV) system running in production use.

I. INTRODUCTION

This paper focuses on the spatial representations of environment required by a mobile robot to cope with everyday activities. Such a representation or map provides means to localize and can be used for planning trajectories between given task coordinates.

In the available literature on mobile robot mapping only a few different environment representations are used. An early example is occupancy grid presented by Moravec and Elfes [11], which provides a probabilistic framework for representing the environment. It is based on the discretization of the environment into cells, which are assumed to be independent from each other. The popularity of the occupancy grid is due to the fact that it is intuitive, easy to implement, and for most purposes efficient for computer programs.

In the mobile robotic mapping it is commonly assumed that the environment is static. The world, however, is dynamic. Changes in the environment are the motivation for this paper and we propose, evaluate and discuss an extension of occupancy grids for dynamic environments called independent Markov chain (iMac) occupancy grid. The proposed extension models each cell as an independent two-state Markov chain. This representation not only describes the probability of a cell being empty or occupied but also provides a prediction of the type of dynamics that can be expected to happen in the cell.

The rest of this paper is organized as follows. Sec. 2 reviews related work on dynamic mapping. Next Sec. 3 introduces the iMac occupancy grid map approach. Sec. 4 describes the test setup and presents and discusses the results. Sec. 5 concludes the paper with discussion on future work.

II. RELATED WORK

There has been significant contributions that have considered localization using static map, but in dynamic environments ([2], [4], [13], [5]). One approach is to change the sensor model so that the expectation of dynamic objects (people) can be incorporated [4]. Another approach is to filter out the measurements that are coming from dynamic obstacles [13], [5]. The above mentioned approaches considers the usage of a static spatial representation and the extensions are added to be able to localize robustly in the dynamic environment, while in our work the goal is to encapsulate the behavior into the representation.

Arbuckle et al. [1] proposes an extension to occupancy grids, which they termed Temporal Occupancy Grids (TOG). TOG is a layered occupancy grid map, where one layer incorporates a certain amount of measurements up to the most recent ones. Each layer thus represents a certain timescale. The disadvantage of the TOG is that the representation needs to preserve the history of measurement equal to the longest timescale. Additionally, the representation needs to be re-computed for each timescale in every iteration. The TOG concept was extended to a Temporal Occupancy Grid (TempOG) by Mitsou and Tzafestas [10]. TempOG represents the occupancy of the space at any time (from the beginning up to the most recent measurement). The representation is based on the Time Index access structure. The representation is an interesting way of capturing the change events in the environment for further analysis, however, it is not efficient because the amount of memory it requires is huge.

Wolf and Sukhatme [14] proposes a method for localization and mapping, which utilizes a static and a dynamic grid map simultaneously. The static map contains information only about static parts of the environment, and the dynamic map contains only information about areas where movement was detected at least once. The information about the static environment is eventually used for localization using an EKF-SLAM. The method is thus trying to identify the dynamic cells in order to avoid erroneous data-association.

Biber and Duckett [3] propose an interesting alternative to represent a dynamic map. Instead of fusing the observation information into any specific representation, they use the observations directly to represent the world. The map proposed in [3] is a set of local maps, organized as “virtual observations”. Each local map thus holds range measurements on a fixed bearing interval as if they would have been recorded from the local map origin. Each bearing interval can incorporate N range measurements. The concept

¹Aalto University, Department of Automation and Systems Technology, PO.Box 15500, FI-00076 Aalto. Jari.Saarinen@aalto.fi

²Center of Applied Autonomous Sensor Systems (AASS), Örebro University, Sweden

of timescale is then introduced by randomly picking M (with $M < N$) samples and replacing them with more recent measurements. In this way the map can represent several timescales. This dynamic map is finally interpreted using robust statistics (median and MAD) in order to avoid values that have not been observed. The disadvantage of the approach in [3] is again the need to store and update large amount of data for the map.

Meyer-Delius et al. [8], [9] propose to use a combination of temporary maps and a static map for localizing in a dynamic environment. The work analyzes parking lot data. Objects are either assumed to be semi-static (cars) or dynamic obstacles, in which case the corresponding readings are filtered out. The position of the robot is updated, if the observation is explained by the static map well enough.

In [9] Meyer-Delius propose to use a state transition probability $p(c_t|c_{t-1})$ to model how the occupancy state changes over time. The assumption is that changes in the environment are due to a stationary process, i.e., $p(c_t|c_{t-1})$ is constant and does not depend on absolute value of t . The state transition probability is estimated using an online (and offline) version of the EM-algorithm. If a particular cell is not observed the state transition probability can be used to predict the cell state. If no observations are made for a long period of time, a cell converges to stationary distribution. The advantage of our approach is simplicity in learning the state transition probabilities. The learning only requires counting of events. Furthermore, the usage of the learned transition parameters differs. In [9] the model describes the amount of change of occupancy in a cell per time unit, which is used as a prior to predict the cell occupancy between observations. In our case the parameters describe the probability of state change event, which is then used to infer the type of dynamics within the cell.

Luber et al. [7] propose to use a spatial affordance map, which is a long-term representation of human-activity events in the environment. Modeling is based on the assumption that observations of moving humans in the environment are independent events and thus observations can be modeled as a nonhomogenous Poisson process. A Poisson process describes the probability to observe a number of events within a given time or volume. Luber et al. [7] approximate the general nonhomogenous Poisson process with a piecewise homogeneous one. Estimation of the rate parameter is done on a grid, where each cell represents a local homogeneous Poisson process with a fixed rate. The Poisson distributed probability of observing k events in a fixed time interval is given by (1):

$$P_{ij}(k) = \frac{e^{-\lambda_{ij}} (\lambda_{ij})^k}{k!}, \quad (1)$$

where ij refers to a particular cell in a grid and λ_{ij} is a rate parameter for the cell. Learning of individual λ_{ij} is carried out as Bayesian estimation with a Gamma distribution prior, which is a conjugate prior of the Poisson distribution. Ultimately, learning of λ_{ij} is a simple additive calculation of the parameters of the Gamma distribution α, β

$$\begin{aligned} \alpha_i &= \alpha_{i-1} + k_i \\ \beta_i &= \beta_{i-1} + 1 \end{aligned}, \quad (2)$$

where k_i is 1 if a moving person was observed in the respective cell, and 0 otherwise. For the estimation of the Poisson rate, the expectation of the Gamma distribution is used:

$$\hat{\lambda} = E[\lambda] = \frac{\alpha}{\beta} = \frac{\#positive\ events + 1}{\#observations + 1}. \quad (3)$$

The interpretation of the rate value given by Eq. 3 put into Eq. 1 with $k = 1$ gives the probability of observing one event given an observation.

Our approach adopts the learning of the Poisson process parameters from [7], however, our extension models two processes, which enables us to use the model not only as general indication of activity, but also to model 1) the occupancy of the cell and 2) to infer the type of dynamics that is expected to happen within the cell.

III. APPROACH

A. The model

We will now formalize our model in a cell level, that is, we will assume that the overall map is divided into independent cells, each of which describes the behavior within that cell. Extending the idea in [7], a map cell is either observed to be occupied or free. Intuitively the process of turning from free to occupied and from occupied to free differs when observing cells with different dynamic properties. As an example, if we are to record people moving in an office space, one would assume to measure most of the time a cell free, until someone appears in the cell. Moreover, given that the person is moving, one would expect that the cell would turn quickly back to free. So, as an extension to Eq. 3 we propose to learn two processes for each cell in the environment:

$$\begin{aligned} \hat{\lambda}_{exit} &= \frac{\alpha_{exit}}{\beta_{exit}} = \frac{\#events: occupied\ to\ free + 1}{\#observations\ when\ occupied + 1} \\ \hat{\lambda}_{entry} &= \frac{\alpha_{entry}}{\beta_{entry}} = \frac{\#events: free\ to\ occupied + 1}{\#observations\ when\ free + 1} \end{aligned}, \quad (4)$$

where α_{exit} is the number of times a cell is observed turning from occupied to free, β_{exit} is the number of observations done in occupied state and α_{entry} and β_{entry} are the respective quantities for observing a cell turning from free to occupied and observing the cell in free state. The additional “+1” in Eq. 4 follows from the initialization of all the parameters to one. The interpretation of $\hat{\lambda}_{exit}$ as a Poisson rate parameter is the expected number of state change events per observation, given that we are in occupied state. Now, we will make an assumption that the rate parameter gives us (an estimate of) a conditional probability of the state change, that is

$$\begin{aligned} \hat{\lambda}_{exit} &\sim p(m = 0 | m = 1) \\ \hat{\lambda}_{entry} &\sim p(m = 1 | m = 0) \end{aligned}, \quad (5)$$

where m indicates a cell state in a given cell. Given Eq. 5 we can derive the state transition matrix for a Markov chain presented in Figure 1 as:

Fig. 1. The model of a map cell is a two state Markov chain

TABLE I
HEURISTIC INTERPRETATION OF THE MODEL

Functional state	λ_{exit}	λ_{entry}
Static occupied	$\rightarrow 0$	High
Static free	High	$\rightarrow 0$
Semi-static	Low	Low
Dynamic	High	Low
Semi-static occupied (doors)	Low	High

$$P = \begin{bmatrix} 1 - \hat{\lambda}_{entry} & \hat{\lambda}_{entry} \\ \hat{\lambda}_{exit} & 1 - \hat{\lambda}_{exit} \end{bmatrix}. \quad (6)$$

Eq. 6 describes the behavior of the cell and gives us the tools to work with, but first let us investigate the expected behavior of the parameters. Table I summarizes the heuristic interpretation of the model. For a static occupied cell, the λ_{exit} tends towards zero as the number of observations increases. Meanwhile on the opposite side, as there are no observations done on the free state, the λ_{entry} stays high (and if never observed the value is one, see Eq. 3). The opposite applies for the static empty. In static cases the chain moves towards the definition of being absorbing chain (however, the chain is regular). On the other extreme, if one would observe a change every time, then both λ_{exit} and λ_{entry} would be one and the chain is periodic. On the semi-static case, it is expected that the cell has a low frequency of changing the state, which implies that there are a lot of observations in both states compared to the state change events. A cell is considered dynamic (meaning that the change events are caused by moving objects) if it is expected that it is mostly empty, but once occupied it turns quickly back to empty. The last entry of Table I is a special case representing, e.g., a door. This is one of the rare cases that one would expect the cell(s) to be mostly occupied, but once free they turn quickly back to occupied.

More formally the behavior of the Markov chain can be studied using some well established properties of it. Given a probability distribution of initial state $u = (u_1, u_2)$, where u_1 is the initial probability of chain being in free state and u_2 is respectively probability of occupied state, the probability distribution after n steps can be computed as:

$$u^{(n)} = uP^n. \quad (7)$$

Markov chain also has a limiting matrix W which is defined as:

$$W = \lim_{n \rightarrow \infty} P^n. \quad (8)$$

Eq. 8 expresses that P converges to a stationary state after sufficient number of steps. For the limiting matrix it applies that for any probability row vector v ,

$$vW = w, \quad (9)$$

where w is the common row vector of W . Furthermore, it applies that $wP = w$, which enables us to analytically compute the stationary probability vector $\pi = (\frac{\lambda_{exit}}{\lambda_{entry} + \lambda_{exit}}, \frac{\lambda_{entry}}{\lambda_{entry} + \lambda_{exit}})$, by solving

$$(w_1, w_2) \begin{bmatrix} (1 - \lambda_{entry}) & \lambda_{entry} \\ \lambda_{exit} & (1 - \lambda_{exit}) \end{bmatrix} = (w_1, w_2). \quad (10)$$

The stationary distribution represents the probability of observing the chain in a particular state given infinite number of steps. The stationary distribution can be used to measure if a cell in a long run would have tendency to be free or occupied, however it is not a good measure of occupancy of the cell. As an example, let us assume that we have a chain

$$P = \begin{bmatrix} 0.999 & 0.001 \\ 0.0001 & 0.9999 \end{bmatrix}.$$

This chain has very high probability of stay occupied if occupied and free if free (representing a semi-static cell). The stationary distribution of this chain is $\pi = (0.09 \ 0.91)$, meaning that in long run the cell is 0.91% occupied, but clearly the cell cannot be considered as static occupied. Instead, we choose Eq. 11 as a measure of static occupancy (and free) the distribution

$$u_o = (0.5, 0.5)P. \quad (11)$$

Eq. 11 returns high probability for the state which is clearly dominant, while for our example it returns $u_o = (0.4995, 0.50045)$, meaning that it is unknown if the cell should be considered free or occupied.

An interesting question is how to reveal different timescales of the model? In [6] and [12] it is shown that for a two state Markov chain, setting initial distribution to $u = (1, 0)$, the first entry of the distribution after k steps is given as

$$u_k(0) = \frac{\lambda_{exit}}{\lambda_{entry} + \lambda_{exit}} + \left(\frac{\lambda_{entry}}{\lambda_{entry} + \lambda_{exit}}\right)(1 - \lambda_{entry} - \lambda_{exit})^k, \quad (12)$$

and the second entry is $u_k(1) = 1 - u_k(0)$. Moreover, given π the distance between the static distribution and u_k is given as:

$$\|u_k - \pi\| = \left(1 - \frac{\lambda_{exit}}{\lambda_{entry} + \lambda_{exit}}\right)|1 - \lambda_{entry} - \lambda_{exit}|^k. \quad (13)$$

The term $|1 - \lambda_{entry} - \lambda_{exit}|$ in Eq. 13 determines the convergence rate. The only cases when Eq. 13 does not converge is when the both transition probabilities equal

to exactly one (periodic case) or zero (absorbing chain with respect to both states). In all other cases the chain will converge to a stationary distribution and the rate is determined by the transition probabilities. Looking at Table I we can expect that for the static cells the convergence should be fast (other λ should be around one), for the dynamic cells the convergence should also be fast and the slowest convergence will happen with semi-static cells.

The exact value of k in Eq. 13 can be straightforwardly computed as

$$k = \frac{\ln(\epsilon \frac{\lambda_{entry}}{\lambda_{exit} + \lambda_{entry}})}{\ln(|1 - \lambda_{exit} - \lambda_{entry}|)}, \quad (14)$$

where ϵ is the desired distance between u_k and the stationary distribution.

B. Recency weighting

At last we will shortly describe a method for adapting to the time variant behavior of the cell. As one can imagine, a cell might change its functional state unexpectedly, which will violate the learned model (for example from static empty to static occupied) and therefore there should be a way to adapt to this change. One such approach is to use recency weighted averaging for the problem (as proposed e.g. in [3]). In our case we have two states that the process can be in and the observations are only made for the one that is active. Our proposal is to limit the maximum number of observations in active state and use forgetting factor for the inactive one. Let $\lambda_i = \frac{\alpha_i}{\beta_i}$ represent the parameter of active state that is updated according to Eq. 2 at time i . Now, if after the update $\beta_{i+1} > N_{limit}$, where the N_{limit} is desired observation limit, then the update is

$$\begin{aligned} \beta_{i+1} &\rightarrow N_{limit} \\ \alpha_{i+1} &\rightarrow \alpha_{i+1} \frac{N_{limit}}{\beta_{i+1}} \end{aligned} \quad (15)$$

In other words, the β_{i+1} is scaled to limit and α_{i+1} is scaled accordingly, so that the value of λ remains identical, but if a new event is observed, it will have higher weight than old ones. Similarly, for the inactive state let now $\lambda_i = \frac{\alpha_i}{\beta_i}$ represent the parameters of inactive state at time i . The update of inactive state is done according to

$$\lambda_{i+1} = \frac{\alpha_{i+1}}{\beta_{i+1}} = \frac{1 + (\alpha_i - 1)\eta}{1 + (\beta_i - 1)\eta}, \quad (16)$$

where η is a forgetting factor. The closer the value η is to 1, the slower is the forgetting.

C. Implementation

The implementation of our model requires storing of:

- Number of exit events
- Number of entry events
- Number of observations in occupied state
- Number of observations in free state
- The state of the last observation (as occupied or free)

In this paper we consider the model to be used with a 2D laser range finder (although the approach is not limited to 2D

nor to be used only with laser). The update simply follows the standard ray-tracing approach and the update is done based on the previous state and the observation according to Eq. 2 for particular state (if the previous state is occupied then only those parameters are updated and vice versa).

In principle one does not have to carry information about both exit and entry events, since $|\alpha_{exit} - \alpha_{entry}| \leq 1$ in all conditions (there cannot be more than two entry, without having exit in between). However, at least we experienced that due to the noise in position the basic implementation causes plenty of events on the borders of the obstacles. Therefore, to smooth this effect we implemented the the update with effective event. This was implemented as $\alpha_i \sim e^{-\frac{d^2}{\sigma^2}}$, where d is the shortest distance to cell with same state as previous observation (within threshold of two times the grid resolution and $\sigma = 0.1m$).

IV. TESTS AND RESULTS

A. Test Setup and environment

To test our approach we collected data from a milk production plant, using a Laser Guided Vehicle (LGV) (see Figure 2) in production use. The data collected includes laser range finder data from a SICK safety laser and position information from the vehicles own navigation computer, thus providing a continuous ground truth information. The data set used to generate the pictures in this paper covers more than ten hours of continuous operation, in total more than 570 thousands of synchronized data points and 8.8km distance covered by the vehicle.

The environment consists of a production area (a corridor shape in the bottom area of Figure 3), and a storage and order picking area (bottom of Figure 2). The task of the LGV is to get an order somewhere from the production area and deliver it to the storage area.

The type of dynamics that one expects in the particular warehouse consists of other LGV's (in total of ten), manually operated forklifts, people and ever changing storage layout (as one may observe from Figure 2 the storage area has little static structures). Additionally, since the data is collected using LGV's own safety laser, there is plenty of reflections that are caused by the floor. Measurements in the experiments are used unfiltered, however, only distances up to 15m are counted to reduce the effect of floor reflections.

B. Evolution model parameters

Figure 4 illustrates the distributions of transition probabilities in different stages of mapping. Even gray area in the images means that there is less than 100 observations done in that state and otherwise the color coding is from 0 to 1 mapped from white to black. On the top there are the values of λ_{entry} . These three images are scaled on a logarithmic scale, since the differences on λ_{entry} are small. This is because λ_{entry} describes the process of turning from empty to occupied, thus mostly observed as free and the values tend to go towards zero. The bottom of Figure 4 represents the evolution of λ_{exit} distributions. In these images it is clearly visible how the busy corridors obtain higher values as the

Fig. 2. The test vehicle (top) and environment (bottom). The data was logged from the SICK safety laser and the reference position was recorded from the LGV's own navigation system.

Fig. 3. A grid map illustrating the test area.

time goes forward. It can be judged that λ_{exit} is a good descriptor for the dynamic cells.

Figure 5 shows the distribution of λ_{entry} , λ_{exit} pairs after the mapping. The figure is a color coded 2D histogram, and the frequency axis is on a logarithmic scale (otherwise the static empty will dominate the figure). The top left corner of the figure represents the static-empty cells in the environment, and bottom right-corner the static occupied ones. The bottom left corner are the semi-static cells and the rest indicate some sort of dynamic cells. The entries of Table I are also mapped in to Figure 5. The figure illustrates that the majority of the cells have low value for λ_{entry} , which was expected. The borders of different functional cells is relatively difficult to estimate from the figure, which is due

Fig. 4. Evolution of the distributions of λ_{entry} (top) and λ_{exit} (bottom) after the experiment. The pictures are from left to right, after inserting 20, 100 and 250 thousands of measurements into the map.

to the sensor (and position) noise. However, there are clearly cells, which behave as described in Table I. Additionally, there is an entry, which is classified as sensor noise (both λ_{entry} and λ_{exit} are high).

C. Estimating occupancy

Figure 6 illustrates the state of the map in the storage area after the experiment. The top figure is computed according to Eq. 11, but using the current measurement to set the prior of u (e.g. if the last measurement for the cell has been occupied then $u = (0.2, 0.8)$). The middle picture is standard occupancy grid and the bottom one is obtained according to Eq. 11 (with inverted colors). The top picture and the occupancy grid seem to agree on the current state, with the difference that on the top picture there some more unknown areas. This is because we have a threshold of 100 observations for a cell before “the content” of it is revealed. More interesting is the bottom image, which should have now only the expected static cells. Comparing against the occupancy grid it can be noticed that the bottom image has a lot more empty space within the storage area. However, there

Fig. 5. Color coded histogram of λ_{entry} , λ_{exit} pairs. The scale is logarithmic.

clearly are also plenty of milk wagons visible that definitely are not static, but this is due to the fact that these places stayed occupied throughout this experiment. This is further elaborated in Figure 7, where the dataset extends over two different days (in total about 21h of data). These pictures represent the static map, but the values have been threshold so that if the probability of occupancy is less than 0.6 then the value is set to white. In these images the storage area is almost empty. Another interesting difference between the occupancy grid and static map in Figure 6 is that the static map looks noise free when observing the static walls.

Figure 7 shows the difference between a static map created with recency weighting (top) and one without recency weighting (bottom). The recency values used were 10,000 for observation limit and forgetting factor of $w = \frac{2000}{2001}$. There is little difference between the top and bottom pictures. The major difference is that some roller cages, which were rendered as free without using the recency weighting have obtained static state with it. This is an expected result, as some of these cages are static in the environment for a long time.

D. Analyzing the timescales

Finally, Figure 8 reveals the “hidden information” of the chains. These figures have been obtained by using different values on N for the Eq. 7, exploiting the property given by Eq. 8 and utilizing the binary Bayesian filter in a following manner: After computing Eq. 7, we compute two distributions $u_1 = (1, 0)P^N$ and $u_2 = (0, 1)P^N$. Knowing that a semi-static cell tends to stay in a state where it is (for small N at least) we use as evidence of semi-static cell $p_1 = u_1(0)$ and $p_2 = u_2(1)$, which are then fused together with Bayesian binary filter. Now, since after the convergence to limiting matrix W , applying Eq. 9 we obtain $u_1 = u_2 = w$, from which it follows that $p_2 = 1 - p_1$ and the filter returns 0.5. So, as long as the chain has not converged, the filter returns the amount of activity that can still be expected on the timescale. It should also be noted that cells which are considered static converge faster than semi-static ones (see Eq. 13).

In Figure 8 the left-most image is taken with $N=8$ and it

Fig. 6. Comparison map states in storage area after experiment. Top: An occupancy map using our approach, middle: standard occupancy grid, and on the bottom: static map given by our approach.

Fig. 7. Comparison of static map with (top) and without (bottom) recency weighting

Fig. 8. Comparison of different timescales, from left to right N equals 8, 16, 32. The more red the area, the higher the activity.

shows a large amount of activity, which is due to the motion and sensor noise, the middle picture is taken with $N=16$, and the right most is with $N=32$. Now, the right most picture starts to reveal the semi-static parts. One can observe from this image the places where the LGV's are waiting for new orders (or others to pass by), as well as doors between the production area and warehouse and roller cages that have been moved during the experiment.

V. CONCLUSION AND FUTURE WORK

In this paper we have introduced a new representation for mapping sensor data into environment model that is based on Markov chain. We have presented a method for learning its parameters and we have shown that it is capable of representing the occupancy of the environment much like occupancy grid, but with much richer representation. For testing we used real world data to verify that the approach does what it promises. We also introduced a way to interpret the timescales using our model. We acknowledge that the model is an approximation, which describes average dynamics on a cell level. At the present formulation the model is also based on an assumption that the dynamics is due to the homogeneous process while in most cases the dynamics tends to have time dependency (the dynamics during the night time differs from the one during the day time). One interesting future work will be to investigate the cyclic behavior of the dynamics by introducing several models, that will be active for certain time periods. Another interesting way for continuation is to study the use of the model for tracking of dynamic objects in the environment by letting the cells communicate among each others (learning the rates of objects moving from one cell to another).

So far in this paper we have not addressed the use cases for the proposed model mostly because this is part of our future work. The obvious use case is to apply the method in a similar way as presented in [7] for tracking applications. The model provides a basis for enhanced localization as well, since the model parameters can be used to influence the likelihood function e.g. by giving higher weight for the observations obtained from likely static structures. Moreover, for planning the model provides a way to penalize the trajectories that are likely to move through high dynamic regions.

REFERENCES

- [1] D. Arbuckle, A. Howard, and M. Mataric. Temporal occupancy grids: a method for classifying the spatio-temporal properties of the environment. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002., 1:409–414, 2002.
- [2] O. Aycard, P. Laroche, and F. Chappillet. Mobile robot localization in dynamic environments using places recognition. In *Proceedings of IEEE International Conference on Robotics and Automation*, 1998., 4:3135–3140, 1998.
- [3] P. Biber and T. Duckett. Dynamic maps for long-term operation of mobile service robots. In *Proceedings of Robotics: Science and Systems (RSS)*, pages 17–24, 2005.
- [4] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *J. Artif. Intell. Res. (JAIR)*, 11:391–427, 1999.
- [5] D. Hahnel, R. Triebel, W. Burgard, and S. Thrun. Map building with mobile robots in dynamic environments. In *IEEE International Conference on Robotics and Automation*, 2003. *Proceedings. ICRA'03.*, volume 2, pages 1557–1563. IEEE, 2003.
- [6] P.G. Hoel, S.C. Port, and C.J. Stone. Introduction to stochastic processes, 1972.
- [7] M. Luber, G. Diego Tipaldi, and K.O. Arras. Place-dependent people tracking. *The International Journal of Robotics Research*, 30(3):280, 2011.
- [8] D. Meyer-Delius, J. Hess, G. Grisetti, and W. Burgard. Temporary maps for robust localization in semi-static environments. In *Intelligent Robots and Systems (IROS)*, 2010 *IEEE/RSJ International Conference on*, pages 5750–5755, oct. 2010.
- [9] Daniel Meyer-Delius Di Vasto. *Probabilistic Modeling of Dynamic Environments for Mobile Robots*. PhD thesis, 2011.
- [10] N.C. Mitsou and C.S. Tzafestas. Temporal occupancy grid for mobile robot dynamic environment mapping. In *Mediterranean Conference on Control & Automation*, 2007. *MED'07.*, pages 1–8. IEEE, 2007.
- [11] H. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *IEEE International Conference on Robotics and Automation. Proceedings. 1985*, volume 2, pages 116 – 121, mar 1985.
- [12] Jeffrey S. Rosenthal. Convergence rates for markov chains. *SIAM Review*, 37(3):pp. 387–405, 1995.
- [13] C.C. Wang and C. Thorpe. Simultaneous localization and mapping with detection and tracking of moving objects. In *IEEE International Conference on Robotics and Automation*, 2002. *Proceedings. ICRA'02.*, volume 3, pages 2918–2924. IEEE, 2002.
- [14] D.F. Wolf and G.S. Sukhatme. Mobile robot simultaneous localization and mapping in dynamic environments. *Autonomous Robots*, 19(1):53–65, 2005.