

Conditional Transition Maps: Learning Motion Patterns in Dynamic Environments

Tomasz Kucner, Jari Saarinen, Martin Magnusson and Achim J. Lilienthal

Abstract—In this paper we introduce a method for learning motion patterns in dynamic environments. Representations of dynamic environments have recently received an increasing amount of attention in the research community. Understanding dynamic environments is seen as one of the key challenges in order to enable autonomous navigation in real-world scenarios. However, representing the temporal dimension is a challenge yet to be solved. In this paper we introduce a spatial representation, which encapsulates the statistical dynamic behavior observed in the environment. The proposed Conditional Transition Map (CTMap) is a grid-based representation that associates a probability distribution for an object exiting the cell, given its entry direction. The transition parameters are learned from a temporal signal of occupancy on cells by using a local-neighborhood cross-correlation method. In this paper, we introduce the CTMap, the learning approach and present a proof-of-concept method for estimating future paths of dynamic objects, called Conditional Probability Propagation Tree (CPPTree). The evaluation is done using a real-world data-set collected at a busy roundabout.

I. INTRODUCTION

Environment representations, or maps, play an important role in robotics. One of the most influential works to robotics mapping has been the occupancy grid map, introduced by Moravec and Elfes [15]. Occupancy grid maps are probabilistic, grid-based representations of static environments. Occupancy maps have been used in many localization [7] and SLAM [10] implementations, and serve as a basis in an abundance of autonomous navigation applications. However, the static-world assumption made in [15] is a shortcoming and recently the attention has shifted from treating the dynamics as a disturbance (as in e.g. [11], [21], [22]) to treating it as part of the representation. However, representing the temporal dimension in a representative manner, is a challenge yet to be solved.

One category of approaches to represent both the spatial and the temporal features of the environment is to store the occupancy signal over time in the map [1], [14]. The disadvantage of such approach is memory consumption and that the environment is not explicitly modeled. Another category aims to estimate the change of occupancy over time [12], [17], [19]. While these representations are compact, the major shortcoming is that the cells are assumed to be independent of each other and thus the change of occupancy does not affect to the neighboring ones. In reality, a predicted

The authors are with the Centre for Applied Autonomous Sensor Systems (AASS), Örebro University, Sweden {tomasz.kucner, martin.magnusson, achim.lilienthal}@oru.se, jari.saarinen@aalto.fi



Fig. 1. The tests in this paper are conducted using a data-set from a busy roundabout. The test environment provides clear patterns of motions and is used in order to emphasize the functionality of algorithm.

change of occupancy in one cell should affect the neighboring cells as well—because objects do not just disappear into thin air.

In this paper, we introduce an environment representation called Conditional Transition Map (CTMap). The CTMap is a grid-based representation, which models transitions of dynamic objects in the environment. The model is created by learning the probability distribution of an object leaving to a certain neighboring cell, given the cell from which it entered into the current cell. This expressiveness comes at the cost of using 64 parameters per cell (e.g., compared to four parameters used by Saarinen et al. [17]). However, the CTMap is also able to predict the likely motions of dynamic objects based on the learned patterns of dynamics, which is a very important feature for future autonomous navigation applications.

As an example, consider the roundabout shown in Fig. 1. and the corresponding CTMap in Fig. 2. The CTMap was trained from observations over 1.5 hours. The vectors, show the exit directions from the cells. Please note that the vectors are distributed along the areas that are highly dynamic, and closely correspond to the shape of the roads as illustrated in Fig. 1.

Along with the CTMap, we also propose a method for learning the transition parameters. The transition parameters are learned from a temporal signal of occupancy of cells

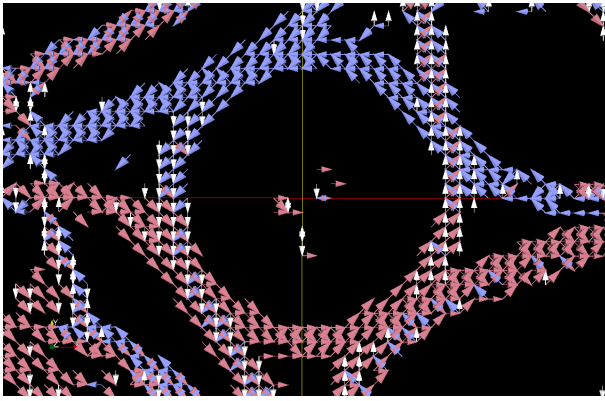


Fig. 2. Pattern of movement on the roundabout, extracted with the described algorithm. As a simple denoising step we have removed the edges with less than 10 exit events. For clarity the entry directions are not shown. (The cell size is 2x2 m. The colors referees to orientation of vectors.)

by using a local-neighborhood cross-correlation method. The learning of the stochastic model only assumes that the motion of the dynamic objects is continuous and that we can observe the motion over a sequence of observations.

The rest of this paper is organized as follows. Sec. II reviews related work, especially on dynamic mapping. Sec. III introduces and formally describes the CTMap, the cross correlation based learning method and the CPPTree. Sec. IV provides an experimental evaluation of the CTMap and CPPTree. Sec. V concludes the paper with discussion.

II. RELATED WORK

Dynamic environments have been considered as a challenge in robotics for a long time. Early approaches treat the dynamics in the environment by altering the sensor model in localization [8] or by removing the dynamic events for mapping [11], [20]. While these approaches have been successful in improving the positioning accuracy to some degree, the stationary-world assumption does not apply if the state of the environment changes essentially.

Arbuckle et al. [1] proposed an extension to occupancy grids. The Temporal Occupancy Grid (TOG) is a layered occupancy grid map, where one layer incorporates a certain amount of measurements up to most recent ones. Arbuckle et al. used the TOG for classifying the dynamics by matching “patterns” (if long term map and medium term map are empty, but the short term map is occupied, then a moving object is observed). The downside of the TOG is that the representation needs to preserve the full history of measurement up to the longest time scale. Instead of fusing observations into any specific representation, Biber and Duckett [2] propose, that observations directly represent the world. The map representation in [2] is a set of local maps, organized as “virtual observations”. Each map has a different learning rate, or timescale, in order to make it possible to both have a map that keeps mostly static features and filters noise, and maps with higher learning rates that more quickly adapt to changes. Using several timescales one can capture the stationary changes in the environment. A related method for learning semi-static states

of the environment was introduced in [3]. The approach extract the map batches where the dynamics have been observed during the different time intervals and learns the different configurations of the batch, which are then utilized to improve the localization. In comparison to our approach, all of the above mentioned approaches represent past states of the environment, however in the level of representation the dynamic behavior of environment is not learned.

Mitsou and Tzafestas in [14] proposed Temporal Occupancy Grid (TempOG), which stores the discrete observations as a time signal for each cell in a grid map. The representation used a time index access structure, which is a special case of a B+ tree. The representation is an interesting approach for capturing the change events in the environment for further analysis. However for on-line use the amount of memory it requires is huge (and grows without bounds as the time goes on). By contrast in our approach we learn parameters of a representation with constant size.

Our approach also relates closely to modeling and prediction of map state changes. Bayesian Occupancy Filters (BOF) [5], [6], [9] have been applied to object tracking. BOF model the environment using a grid representation, with additional velocity components. The occupancy transitions are then predicted using collected observations of moving obstacles. In addition, in [9] the prior knowledge of the shape of the environment to regulate the possible motions in prediction of an object trajectory. In this approach the probabilities of transitions depends on the velocity and geometrical constrains of the environment. Our approach in contrast learns these probability distributions based on observed motion gaining additional implicit knowledge. In fact our representation could be used as an extension for BOF; however, this is out of the scope of this paper.

Luber et al. in [12] proposed to use a spatial affordance map in the context of people tracking. A spatial affordance map is a long-term representation of human-activity events in the environment. The model is grid based and assumes that an observation of a moving human in the environment is an independent event. Events are modeled as nonhomogenous Poisson processes. The spatial affordance map was used to represent the probability of change events and it was used to improve people tracking performance.

Meyer-Delius [13] introduced an expectation-maximization-based approach to learn state-transition probabilities for an occupancy map. The changes in the environment were assumed to be caused by semi-static objects and due to a stationary process. The assumption of semi-static changes was relaxed in [16] by using a similar approach for learning as in [12], but modeling each cell as an independent two-state Markov chain (iMac).

Common for Luber et al. [12], Meyer-Delius [13] and Saarinen et al. [16] is that they learn the patterns of dynamic changes at the cell level, assuming independent cells. In reality, when the occupancy state is predicted to change in some cell, it must affect the neighboring cells, since matter does not disappear suddenly. The CTMap represents conditional transition probabilities for each cell. Thus, in

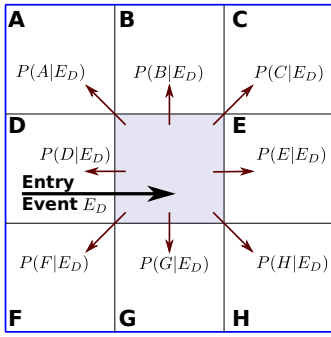


Fig. 3. An example of conditional probabilities in CTMap. The cell describes the probability distribution over all eight exit directions for an event arriving from cell D.

contrast to earlier work, it explicitly models the probability distribution of an object to move from the current cell to neighboring cells. Linking entry and exit events of cells provides a tool to capture and predict motion patterns.

In this paper we propose to learn transition parameters between cells by using a local-neighborhood cross-correlation method. Cross-correlation is a well-known signal processing method.

Our learning method has some similarities to the fast optical flow algorithm [18]. However in our approach we are only focusing on cells or pixels whose state is changing over time and we are working with more than just subsequent frames. Our approach further relates to visual crowd monitoring techniques [23], particularly with the floor field approach [4]. However, while CTMap is focusing on the properties of space, the crowd monitoring algorithms are trying to learn explicitly the behavior of objects.

III. CONDITIONAL TRANSITION MAP

A. The representation

CTMap is a grid-based probabilistic model describes the state-transition process from one cell to its neighbors. Each cell in the grid contains eight sets of eight parameters; in total 64 parameters. Each of the sets is associated with one possible entry direction into the cell. Each set of parameters describes the probability of exit directions, conditioned on the given entry direction. In our approach we are only modeling motion of object, all stationary objects are filtered out. A visualization of CTMap is given in Fig. 3. An object, has arrived from cell D and this constitutes an entry event. The set of arrows pointing outwards represents the conditional exit probabilities.

The parameters of the CTMap are learned by counting frequencies of events over time. Both entry and exit events are detected by computing cross-correlation of the occupancy signal for adjacent cells.

B. Cross-correlation-based learning of parameters

In this paper we assume that objects are moving in 2-D on continuous paths $f(t)$.

The objects will move in one time step only to adjacent cells, given an appropriate discretization of space and time

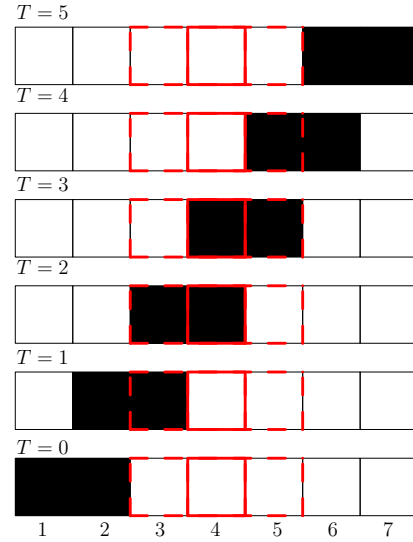


Fig. 4. Sequence of one dimensional maps

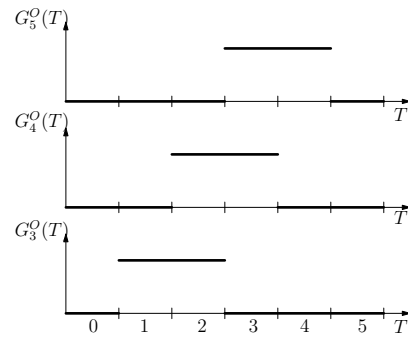


Fig. 5. State of cells 3,4,5 in time

(according to the Nyquist-Shannon theorem and a known maximum speed of objects).

We associate to each cell in the grid a function which describe the state over time:

$$G_{X,Y}^O(T) = \begin{cases} 1 & \text{if occupied} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

A cell is considered occupied if any object was present in the cell during time step, otherwise we mark the cell as empty.

Fig. 4 shows a sequence of one dimensional occupancy maps. We are focusing on the *central cell* (number 4) with *adjacent cells* number 3 and 5. The figure illustrates an object moving from left to right, and Fig. 5 shows the corresponding state change signals for the cells 3, 4, and 5 over time. Fig. 5 show how the occupancy signal is shifted in time for each cell. Treating changes of state as a signal in time we can compute the relative shift parameters using cross-correlation, which for our purpose as:

$$(G_i^O \star G_j^O)[m] = \sum_{m=0}^{2N-T-1} G_i^O[T+m]G_j^O[m]. \quad (2)$$

We assume that each signal is finite and equals to N samples. The number of samples is proportional to the time that the

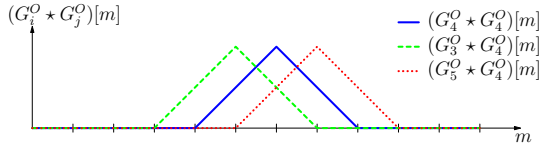


Fig. 6. Graphs of cross-correlations

cell was observed as being occupied. The cross-correlation of the signals in Fig. 5 is shown in Fig. 6.

The next step is to distinguish the shift direction. The shift is computed using following equation:

$$S_{i,j} = \arg \max_m (G_j^O * G_j^O)[m] - \arg \max_m (G_i^O * G_j^O)[m] \quad (3)$$

To determine the direction of movement we need to calculate the $\text{sgn}(S_{i,j})$. If it is equal to 1 the object has entered the cell, if -1 the object has exited the cell and if 0 cell was static. Transition is a single action of moving the object across the edge of the cell. Since we are considering continuous motion, an entry transition is always related to a corresponding exit transition. Each transition is characterized by one peak of $G_{X,Y}^O$ (that is, a state change from low to high, and to low again) in the central cell. To compute the cross correlation of a transition we extract only the time interval where there is simultaneously a peak in G in at least one of the neighboring cells. Fig. 8 illustrates two objects passing through the neighborhood, and the resulting time series for marked cells. The vertical line separates the intervals, which will be analyzed for the movement detection. As we can see, the red object is entering cells earlier than the green one. Both of them are changing the state of each cell, which is visible on the graphs related to each cell. Our method gives proper results if there is only one peak in the analyzed interval, therefore we divided each time series according to that requirement as shown in the Fig. 8.

In most scenarios objects do not move completely randomly and thus the exit direction depends on the entry direction. The CTMap estimates the probability distribution of exit directions conditioned on the entry direction. The parameters of distribution are learned by counting frequencies of events given by Eq. (4). To distinguish the exit directions we use the information provided by cross-correlation. In the future we will replace this with more robust estimator.

$$P(\text{Exit}_i | \text{Enter}_j) = \frac{\#\text{Exits to } i}{\#\text{Enters from } j} \quad (4)$$

Because we are using a discretized grid as underlying representation, there are cases in which the entries and exits are detected in multiple directions simultaneously. We have to assume that each entry and exit is equally possible therefore we get a set of entries and exits. Next we combine all entry directions with all exit directions into tuples (entry,exit) as illustrated in Fig 7 and then we use those tuples to consistently update the frequencies.

All the equations introduced in 1-D can be easily modified for 2-D just by extending the size of the neighborhood from 2 to 8 cells.

The procedure of building the CTMap is as follow:

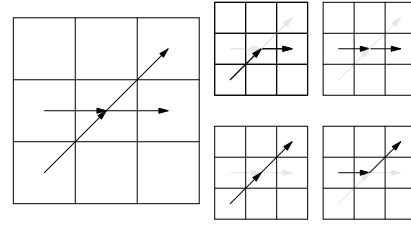


Fig. 7. Set of tuples for a single analyzed time interval

- 1) Creation of occupancy grid-map snapshots - Combine the scans to build a map of environment for every considered time interval.
- 2) Extract time series for each cell - Build a vector of states for each cell in the map in time.
- 3) Analysis of the neighborhood
 - a) Divide time-series into intervals - In Fig 8 it is shown that each interval contain only one event of occupancy for each cell and start and ends at the same moment for all adjacent cells.
 - b) Compute the direction of movement for each neighborhood - See Eq. (3).
- 4) Compute set of conditional probabilities for each cell - See Eq. (4).

C. Conditional Probability Propagation Tree

The Conditional Probability Propagation Tree (CPPTree) is a graph representing all reachable transitions which are connected to the initial transition. Information provided by CPPTree can be used either to estimate the future paths of tracked objects or extract reachable areas for path planning.

Since the CTMap is focusing on the transitions between cells, to initialize CPPTree building algorithm we need two adjacent cells, between which there has been a transition. For a given number of iterations, to each node, all reachable edges are added (if the probability of transition along the edge is higher than a threshold).

IV. RESULTS

In this section we will present and discuss the results of CPPTree and CTMap.

A. Data Set Preparation

The evaluation of the algorithm was done by using a data set recorded during Swedish rush hour with a Velodyne-HDL64 3D laser scanner situated at the center of the roundabout shown in Fig 1. The algorithm requires a time series of occupancy for each cell. The 3D data is projected to 2D by using the ROS (see ros.org) node velodyne_obstacles from the package velodyne_height_map(see http://www.ros.org/wiki/velodyne_height_map, used version 0.3.0.). To get complete map for each interval we combine a sequence of three scans.

B. Model Training

Our current approach records all disturbances and outliers observed during the learning step. Because all neighborhoods

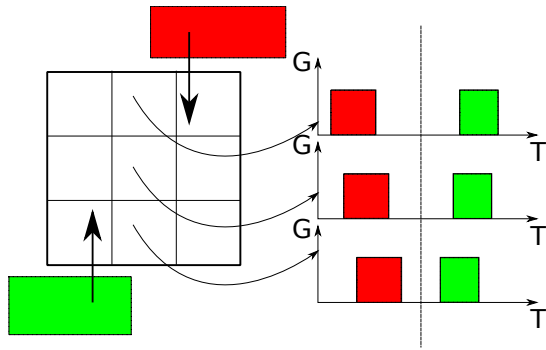


Fig. 8. Intervals over time for a sequence of cells.

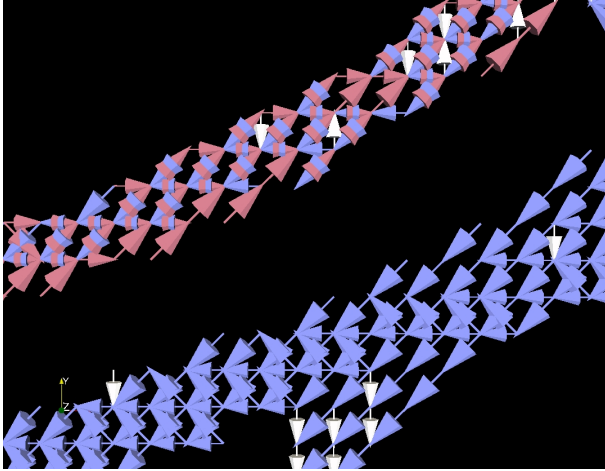


Fig. 9. Pattern of motion for sidewalk(upper one) and road(lower one).

are treated independently, there is also no way to distinguish if the measurement is part of a longer chain of events.

On the other hand, the model keeps detailed information about the environment. As we can see in Fig.9, for example the sidewalk differs from the road. The motion pattern of the road is orientated along one direction, while the vectors on the sidewalk are pointing in all directions. The reason is that on the road all objects are moving in one direction (restricted by the traffic rules), while on side walk people are going in both directions on the same path.

C. CPPTree example

The main purpose of the CPPTree is to build a set of reachable transitions between adjacent cells, starting from a given initial transition. As we can see in Fig. 10 the set of transitions define all the possible paths starting in the root node determined by the initial transition. This graph provides information about all reachable points on the map from the present configuration. In the example considered in Fig. 10, we can see that not only the geometrical constrains were capture but also additional information about the allowed direction of motion. We can exploit this information for object tracking or for path planning.

D. Memory consumption

The CTMap does not only encode the spatial configuration but also temporal changes of the environment. We therefor

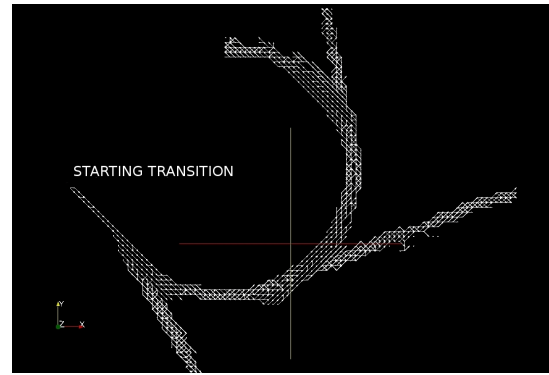


Fig. 10. The CPPTree (cell size 0.8x0.8 m)

compare its memory consumption to approaches that also model temporal changes. Saarinen et al. [17] use less parameter however do not catch the relations among adjacent cells. The approach presented by Arbuckle et al. [1] does not require a fixed number of parameters to represent the dynamics in the environment and the memory consumption rise as the length of the length of the longest interval rises. We expect that CTMap has a lower memory requirements.

V. SUMMARY & FUTURE WORK

In this paper we have introduced CTMap: a probabilistic representation of the dynamics in an environment. CTMap extends a grid map representation with a set of conditional probabilities describing the expected direction of movement. Additionally, an approach for learning the parameters of CTMap was introduced. Finally, in order to demonstrate the expressiveness of CTMap, we introduced the Conditional Probability Propagation Tree, which extracts the predicted path of an observed objects. The evaluation of the approach was done using a real-world data set from a busy roundabout. We observed that the CTMap quickly convergences to a sensible distribution and that the structure remains stable over long periods of time. This data set represented ideal conditions for our algorithm, providing plenty of large dynamic objects that follow traffic rules in the environment. We also note that the dynamics from the pedestrians on a walkway were convincingly mapped by the CTMap resulting in a pattern that identifies the main motion directions. Finally, the CPPTree was shown to be able to predict the expected motion patterns for vehicles.

The CTMap is currently at an early stage and there are many long-term autonomy applications that can greatly benefit from the approach. One of the most promising directions is in the field of object tracking. The CTMap provides an excellent basis for predicting over the whole map. This, on the other hand, opens new revenues for planning safe actions in dynamic environments. Ultimately, an autonomous robot acting among humans should be able to operate in a manner that is the least disturbing or stressful for the humans. Thus, learning the patterns of motions in an environment will also help in planning trajectories that take into a consideration the most likely actions of humans.

There are many possible directions for future work, including:

a) *Incorporating time and velocity*: The present version of CTMap only estimates the probability of transition over the cell edges. A natural next step is to incorporate the time into the model, in order to be able to predict the trajectory of objects, not just the path. One promising approach is to merge CTMap and the iMac approach proposed in [17].

b) *Partial observability*: Although the data used in this paper are only partially observable, artifacts coming from occlusions only had a minor influence on the result; mainly because the whole map area is within the range of the stationary sensor. A mobile sensor platform, will have to deal with effects of partial observability. An important topic for future work will thus be to enable CTMap to function in situations where large parts of the environment are not visible for extended time periods.

c) *Different Scenarios*: The approach was tested so far only in one scenario. A major direction of future work will be to test the performance of the algorithm in different scenarios.

d) *Vector field*: In the present implementation, each transition is treated as an edge with an associated weight. Every cell is associated with 64 such edges, therefore the size of the CPPTree graph grows exponentially, and does not scale well with larger maps. Instead of building a CPPTree graph, another option is to treat the transitions as a vector field, which exerts influence on an object moving through the map.

e) *Dynamic motion patterns*: The CTMap models the general motion patterns in an environment, but in its present form it assumes that the motion patterns themselves are static. In some cases, the motion patterns may change depending on the time of day; for example, with a path being traversed in one direction in the mornings and the opposite direction in the evenings. Future work should also focus on modeling temporal variations of the dynamics.

f) *Semantically meaningful regions*: In the form that we can see in Fig. 9 the map is very cluttered and is very difficult to read. To make the output more human friendly we are going to build a representation which will show clustered regions with consistent exit properties.

ACKNOWLEDGMENT

This work has partly been supported by the European Commission under contract number FP7-ICT-600877 (SPENCER) and the Stiftelsen fr kunskaps- och kompetensutveckling under contract number 20110214 (ALLO).

REFERENCES

- [1] D. Arbuckle, A. Howard, and M. Mataric. Temporal occupancy grids: a method for classifying the spatio-temporal properties of the environment. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, pages 409–414. IEEE, 2002.
- [2] P. Biber and T. Duckett. Dynamic maps for long-term operation of mobile service robots. In *Proc. of Robotics: Science and Systems (RSS)*, pages 17–24, 2005.
- [3] W. Burgard, C. Stachniss, and D. Hähnel. Mobile robot map learning from range data in dynamic environments. *Autonomous Navigation in Dynamic Environments*, pages 3–28, 2007.
- [4] Carsten Burstedde, Kai Klauck, Andreas Schadschneider, and Johannes Zittartz. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A: Statistical Mechanics and its Applications*, 295(3):507525, 2001.
- [5] Cheng Chen, Christopher Tay, C Laugier, and Kamel Mekhnacha. Dynamic environment modeling with gridmap: a multiple-object tracking application. In *Control, Automation, Robotics and Vision, 2006. ICARCV'06. 9th International Conference on*, pages 1–6. IEEE, 2006.
- [6] C Cou, Th Fraichard, Pierre Bessiere, and Emmanuel Mazer. Multi-sensor data fusion using bayesian programming: An automotive application. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 1, pages 141–146. IEEE, 2002.
- [7] Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Monte carlo localization for mobile robots. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 1322–1328. IEEE, 1999.
- [8] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *J. Artif. Intell. Res. (JAIR)*, 11:391–427, 1999.
- [9] Tobias Gindele, Sebastian Brechtel, Joachim Schroder, and Rüdiger Dillmann. Bayesian occupancy grid filter for dynamic environments using prior map knowledge. In *Intelligent Vehicles Symposium, 2009 IEEE*, pages 669–676. IEEE, 2009.
- [10] G. Grisetti, G.D. Tipaldi, C. Stachniss, W. Burgard, and D. Nardi. Fast and accurate slam with rao-blackwellized particle filters. *Robotics and Autonomous Systems*, 55(1):30–38, 2007.
- [11] D. Hahnel, R. Triebel, W. Burgard, and S. Thrun. Map building with mobile robots in dynamic environments. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 2, pages 1557–1563. IEEE, 2003.
- [12] M. Luber, G. Diego Tipaldi, and K.O. Arras. Place-dependent people tracking. *The International Journal of Robotics Research*, 30(3):280, 2011.
- [13] Daniel Meyer-Delius Di Vasto. *Probabilistic Modeling of Dynamic Environments for Mobile Robots*. PhD thesis, 2011.
- [14] N.C. Mitsou and C.S. Tzafestas. Temporal occupancy grid for mobile robot dynamic environment mapping. In *Control & Automation, 2007. MED'07. Mediterranean Conference on*, pages 1–8. IEEE, 2007.
- [15] Hans Moravec and Alberto Elfes. High resolution maps from wide angle sonar. In *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volume 2, page 116121, 1985.
- [16] Jari Saarinen, Henrik Andreasson, and Achim Lilienthal. Markov chain based model of dynamic environment. In *In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems October 7-12, 2012, Vilamoura, Algarve, Portugal, 2012*.
- [17] Jari Saarinen, Henrik Andreasson, and Achim J. Lilienthal. Independent markov chain occupancy grid maps for representation of dynamic environments. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2012*. to appear.
- [18] Changming Sun. Fast optical flow using 3d shortest path techniques. *Image and Vision Computing*, 20(1314):981 – 991, 2002.
- [19] Daniel Vasto Meyer-Delius Di Vasto. *Probabilistic Modeling of Dynamic Environments for Mobile Robots*. PhD, Albert-Ludwigs-Universität Freiburg im Breisgau, Breisgau, August 2011.
- [20] C.C. Wang and C. Thorpe. Simultaneous localization and mapping with detection and tracking of moving objects. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 3, pages 2918–2924. IEEE, 2002.
- [21] Chieh-Chih Wang and Chuck Thorpe. Simultaneous localization and mapping with detection and tracking of moving objects. In *In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, page 842849, 2002.
- [22] Denis F. Wolf and Gaurav S. Sukhatme. Mobile robot simultaneous localization and mapping in dynamic environments. *AUTONOMOUS ROBOTS*, 19:5365, 2005.
- [23] Beibei Zhan, Dorothy Ndedi Monekosso, Paolo Remagnino, Sergio A. Velastin, and Li-Qun Xu. Crowd analysis: a survey. *Mach. Vis. Appl.*, 19(5-6):345–357, 2008.