Deploying Artificial Landmarks to Foster Data Association in Simultaneous Localization and Mapping

Maximilian Beinhofer Henrik Kretzschmar Wolfram Burgard

***Abstract*— Data association is an essential problem in simultaneous localization and mapping. It is hard to solve correctly, especially in ambiguous environments. We consider a scenario where the robot can ease the data association problem by deploying a limited number of uniquely identifiable artificial landmarks along its path and use them afterwards as fixed anchors. Obviously, the choice of the positions where the robot should drop these markers is crucial as poor choices might prevent the robot from establishing accurate data associations. In this paper, we present a novel approach for learning when to drop the landmarks so as to optimize the data association performance. We use Monte Carlo reinforcement learning for computing an optimal policy and apply a statistical convergence test to decide if the policy is converged and the learning process can be stopped. Extensive experiments also carried out with a real robot demonstrate that the data association performance using landmarks deployed according to our learned policies is significantly higher compared to other strategies.**

I. Introduction

One of the fundamental problems in mobile robotics is simultaneous localization and mapping (SLAM). In SLAM, a robot builds a map of an initially unknown environment while localizing itself in this very map. One of the key challenges during SLAM is that of data association, where the robot has to recognize previously observed places. In general, data association failures lead to inconsistent maps that cannot be used for navigation tasks. Whereas highly effective methods for computing a map given the data associations have been developed in the past [6], [9], the development of methods for robust data association is still an open research problem. In practice, data association quickly becomes intractable, particularly in ambiguous environments, as the complexity of the data association problem grows exponentially with the number of feature observations.

One way of resolving ambiguities in the environment and supporting data association is to deploy artificial landmarks. In the past, several approaches to foster potential future data associations have been developed. They include robots that can drop radio-frequency identification (RFID) tags or similar uniquely identifiable landmarks [4], [8], [17]. In all these approaches, however, there is the problem of deciding where and when to deploy the landmarks. This problem becomes even more relevant when the robot can only carry a limited number of landmarks.

All authors are with the Department of Computer Science at the University of Freiburg, Germany. This work has partly been supported by the German Research Foundation (DFG) within the Research Training Group 1103 and under contract number SFB/TR 8, and by the EC under contract number FP7-600890-ROVINA. The authors thank Rainer Kümmerle and Dominik Joho for their support regarding g2o and the RFID reader, respectively. beinhofe@informatik.uni-freiburg.de

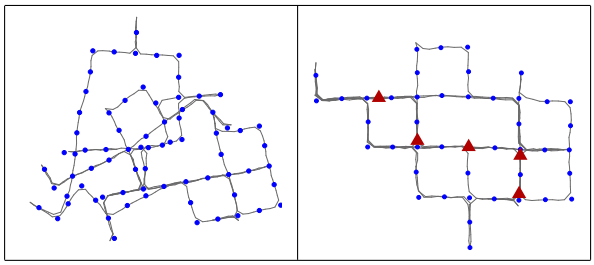

Fig. 1. Estimated landmark positions (blue circles) and robot path (gray line) for a simulated robot traveling through a Manhattan-like world with data associations calculated without the use of deployed markers (left) and with the uniquely identifiable markers (red triangles) deployed by our approach (right).

In this paper, we present an approach for learning a policy for autonomous landmark deployment that aims at optimizing the data association performance. Our method is designed to assist the SLAM system without interfering with the actual navigation tasks carried out by the robot. Consequently, the robot does not need to perform any detours for proper landmark deployment. To compute the optimal policy, we apply actor-critic Monte Carlo reinforcement learning using the number of incorrectly estimated feature correspondences as performance measure. For learning, we employ simulated episodes of robot navigation tasks. In order to make the resulting policies generalize well to different environments, our approach relies on general features like the remaining battery life time, the number of landmarks left on board, the distance to the closest deployed landmark, and a feature capturing the abstract local structure of the environment. To reduce the number of episodes required for learning, we employ a statistical convergence test. As a result, the robot can efficiently learn a policy for placing artificial landmarks so that the data association errors are greatly reduced. Fig. 1 provides an illustrative example for a synthetic environment with non-distinguishable landmarks.

This paper is organized as follows. After discussing related work in the next section, we give a short survey on SLAM and reinforcement learning in Section III. In Section IV, we introduce our approach for learning artificial landmark deployment policies. Finally, we provide extensive experiments that demonstrate the effectiveness of the learned policies both in simulation and on a real robot.

II. Related Work

In the past, several approaches to tackle the data association problem in SLAM have been developed. One popular method that does not rely on artificial landmarks

is the joint compatibility branch and bound method by Neira and Tardós [12]. It explicitly considers the correlations between landmarks by searching an interpretation tree for the hypothesis that covers the largest number of jointly compatible pairings. Olson [13] looks for local matches in the environment and aims to reject those matches that are not globally consistent using single cluster graph partitioning, which relies on a pair-wise consistency graph. In contrast to our approach, methods without the aid of artificial landmarks have to solely rely on the landmarks present in the environment. Therefore, with increasing ambiguity in the environment it becomes more challenging for such methods to robustly find the correct data associations.

The majority of approaches for SLAM with the aid of deployable landmarks address graph-like worlds and deterministically observable markers. For example, Dudek *et al.* [4] localize a robot that travels along the edges of a graph and that can deploy and identify markers at the vertices. Bender *et al.* [2] present approaches for mapping a directed graph using deterministically observable undirected markers. Wang *et al.* [17] prove that the SLAM problem in an undirected graph can be solved deterministically if the robot can drop a deterministically observable directional marker. In contrast to these approaches, we apply a probabilistic model to deal with noisy motion and measurements.

Batalin and Sukhatme [1] devised a coverage strategy for a robot with no knowledge about its position. In their case, the robot can deploy active markers and use them later to move into the direction suggested by them. In the work by Kleiner *et al.* [8], the robot applies a manually designed heuristic, which takes into account the obstacle density and the estimated tag density, to deploy RFID markers to aid a SLAM system. In contrast to such approaches, our method learns a landmark deployment policy from simulated runs.

The problem of selecting informative environment features in SLAM is closely related to the problem considered in this paper. For example, Strasdat *et al.* [14] use reinforcement learning to determine a policy for feature selection which minimizes the distance between the final position of the robot and its goal. They consider obstacle-free worlds and therefore do not need to incorporate information about the spatial structure of the environment into the learning method. Thrun [16] considers a similar problem in the context of mobile robot localization and uses the average posterior localization error for deciding which features to use.

Compared to previous methods, the contribution of the work presented in this paper is a novel approach to improve data association performance by applying a policy for autonomous deployment of artificial landmarks, learned using actor-critic Monte Carlo reinforcement learning.

III. Background

A. Simultaneous Localization and Mapping

Simultaneous localization and mapping (SLAM) refers to the problem of estimating the joint posterior distribution

$$p(x_{1:T}, m_{1:n}, c_{1:T} \mid u_{1:T}, z_{1:T}) \tag{1}$$

of the robot's poses $x_{1:T}$ and the map m, which consists of n features $m_{1:n}$, given a set of robot motion commands $u_{1:T}$ and a set of feature observations $z_{1:T}$. Furthermore, $c_{1:T}$ refers to the data associations, i. e., the identities of the map features perceived in the observations $z_{1:T}$.

B. Data Association in SLAM

In the context of the simultaneous localization and mapping problem, data association refers to the problem of identifying a map feature m_i in one observation z_{t_1} as the very same feature found in another observation z_{t_2}. Unless the robot is able to recognize previously visited places, its position uncertainty increases without bound due to the accumulating odometry error. Integrating out the unknown data associations $c_{1:T}$ in Eq. (1) leads to

$$\begin{aligned} &p(x_{1:T}, m_{1:n} \mid u_{1:T}, z_{1:T}) \\ &\quad = \sum_{c_1} \sum_{c_2} \dots \sum_{c_T} p(x_{1:T}, m_{1:n}, c_{1:T} \mid u_{1:T}, z_{1:T}) \,. \end{aligned} \tag{2}$$

Consequently, the number of possible data associations grows exponentially with the number of observations.

Most approaches to data association are based on the innovation and its covariance, i. e., the difference between the actual observation z_i and the predicted observation $\hat{z}_i$ under a given data association $c(z_i)$. If the innovation is a Gaussian, the squared Mahalanobis distance $D_M^2(z_i, \hat{z}_i)$ is distributed according to the χ_d^2 distribution, where d is the dimensionality of the innovation. One of the most popular approaches to data association is the nearest neighbor filter. It computes a set of compatible candidate features and accepts only features whose innovation is within a certain region of the χ^2 distribution. It then chooses the candidate feature that best matches the observation.

There are more sophisticated data association techniques than the nearest neighbor filter [12], [13], which can handle significantly more challenging environments. However, even these approaches cannot guarantee to avoid false positives, particularly in the presence of perceptual ambiguities.

C. Graph-Based Approaches to Solve SLAM

Graph-based approaches to solve the SLAM problem model the poses of the robot and the positions of observed landmarks as nodes in a graph. The edges of such a graph correspond to spatial constraints between the individual nodes. These constraints arise from odometry measurements and from landmark observations. Graph-based SLAM approaches are typically divided into a front end and a back end. The front end interprets the sensor data to extract spatial constraints. To do so, the data association problem has to be addressed. Using the solution of the data association problem, the back end finds the configuration of the nodes that best matches the extracted spatial constraints by applying an optimization technique [6], [7], [9]. In the back end, a key precondition of the successful computation of the map is getting the correct data associations from the front end.

In our experiments, we apply a graph-based SLAM approach using a nearest neighbor filter for data association in

the front end and the optimization from [9] in the back end. Note that our approach is not restricted to this framework but can be applied in any SLAM system.

D. Actor-Critic Monte Carlo Reinforcement Learning

In reinforcement learning [15], an agent interacts with its environment to learn how to behave so as to maximize a numerical reward. Formally, a reinforcement learning problem is given by a set of states $\mathcal{S}$, a set of actions $\mathcal{A}$ that the agent may perform at discrete time steps t, transition probabilities that describe how the states change in response to the agent's actions, and a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ that determines the numerical reward that the agent receives for executing action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$. A policy is a probability distribution $\pi(s, a) = p(a \mid s)$ of choosing action a given the agent is in state s. For each episode e, which is a sequence $(s_0, a_0, \ldots, s_T, a_T)$ of states and actions, the return R_t for executing action a_t is given by the sum of the rewards gathered after the execution of a_t:

$$R_t = \sum_{t'=t+1}^{T} r(s_{t'}, a_{t'}) \,. \tag{3}$$

The goal of reinforcement learning is to find the policy $\pi^\star(s, a)$ that maximizes the expected return

$$Q^\pi(s, a) = \mathbb{E}_\pi[R_t \mid s_t = s, a_t = a] \tag{4}$$

for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$.

Monte Carlo reinforcement learning methods do not require prior knowledge of the environment's dynamics, i. e., the probability distributions of the transitions, to compute the optimal policy. Instead, these methods estimate $Q^\pi(s, a)$ using the return averaged over a number of sample episodes e. First-visit-only Monte Carlo learning takes into account only the first occurrence of each state-action pair (s, a) in each episode e, leading to the estimator

$$\hat{Q}^\pi(s, a) = \frac{1}{n_\mathcal{F}} \sum_{e \in \mathcal{F}(s,a)} R^e_{\text{first}}(s, a) \,, \tag{5}$$

where $\mathcal{F}(s, a) = \{e \mid (s, a) \in e\}$, $n_\mathcal{F} = |\mathcal{F}(s, a)|$, and $R^e_{\text{first}}(s, a)$ is the return at the first occurrence of (s, a) in episode e. In Monte Carlo reinforcement learning, the estimator $\hat{Q}^\pi$ converges to Q^π if all state-action pairs occur with non-zero probability when following the policy π. A common way to satisfy this condition is to use a so-called softmax policy of the form

$$\pi(s, a) = p(a \mid s) = \frac{\exp(\hat{Q}^\pi(s, a)/\tau)}{\sum_{a' \in \mathcal{A}} \exp(\hat{Q}^\pi(s, a')/\tau)} \,, \tag{6}$$

where τ is the so-called temperature.

In actor-critic reinforcement learning, the actor follows a policy, while the critic attempts to estimate the Q-function under this policy. As soon as the critic has observed enough episodes to learn the Q-function, the critic becomes the actor and a new critic is initialized. Actor-critic learning thus does not change the policy while learning the Q-function. As a result, the estimate of the Q-function is not distorted by values induced by a policy that has already been altered.

IV. Our Approach to Deploying Artificial Landmarks to Foster Data Association

To facilitate the data association during SLAM, we consider a robotic system that can autonomously deploy a limited number k of uniquely identifiable landmarks. Given these additional landmarks, the SLAM posterior turns into

$$p(x_{1:T}, m_{1:n}, c_{1:T}, l_{1:k} \mid u_{1:T}, z_{1:T}, z^l_{1:T}, c^l_{1:T}) \,, \tag{7}$$

where $l_{1:k}$ are the positions of the artificial landmarks, and $c^l_{1:T}$ are the known identities of these landmarks perceived in the observations $z^l_{1:T}$. The observations $z^l_{1:T}$ of the deployed artificial landmarks and, in particular, the correspondences $c^l_{1:T}$ refine the SLAM posterior, potentially making the data association problem more tractable by resolving ambiguities in the environment.

The benefit of the artificial landmarks for data association obviously depends on where and when the robot deploys them. We aim at distributing the artificial landmarks such that the risk of wrong data associations for the environment features is minimized. Note that our method is designed to assist the SLAM system of the robot without interfering with the actual navigation task carried out by the robot. Our approach deploys the artificial landmarks along the trajectory of the robot and does not impose detours to deploy artificial landmarks at appropriate positions. Our approach applies Monte Carlo reinforcement learning to compute a policy that allows the robot to deploy the artificial landmarks such that the performance of data association is optimized.

A. Measuring the Performance of Data Association

To measure the performance of data association, we count the number of incorrectly estimated map feature correspondences. Let $c^\star_t$ be the true data association that indicates that observation z_t stems from environment feature $m^\star_i$. In contrast to that, c_t is the correspondence of observation z_t to a map feature m_j as estimated by the data association method. For every environment feature $m^\star_i$, we count the number $N(m^\star_i)$ of map features m_j that the data association method associated at least once with $m^\star_i$. More formally, we define

$$N(m^\star_i) = |\{m_j \in m_{1:n} \mid \exists t : c^\star_t = m^\star_i \wedge c_t = m_j\}| \,. \tag{8}$$

If the feature correspondences are correctly estimated, we have $N(m^\star_i) = 1$. Accordingly, the total number of incorrectly estimated feature correspondences is given by

$$E(c^\star_{1:T}, c_{1:T}) = \sum_{m^\star_i} (N(m^\star_i) - 1) \,. \tag{9}$$

Our approach aims at placing the artificial landmarks such that the number of incorrectly estimated feature correspondences E is minimized.

B. Reinforcement Learning for Improving Data Association

Extensive experiments (see Sec. V) revealed that heuristics for deciding when to deploy landmarks perform badly or need to be hand-tuned for specific scenarios. Therefore, we apply actor-critic Monte Carlo reinforcement learning as

described in Sec. III-D to estimate a landmark deployment policy. We compute a policy that allows the robot to deploy a set of artificial landmarks at the locations that minimize the risk of wrong data associations in terms of the error E defined in Eq. (9). In each simulated episode, the robot performs a randomly sampled navigation task. The robot thereby applies graph-based SLAM and deploys its artificial landmarks according to the currently estimated policy. In each of the episodes, the robot receives the rewards

$$r_t = \begin{cases} 0 & \text{if } t < T, \\ -E(c^{\star}_{1:T}, c_{1:T}) & \text{if } t = T. \end{cases} \tag{10}$$

C. Action and State Representation

At every time step t, the robot decides whether to drop one of the artificial landmarks in the current state s according to a policy $\pi(s, a)$. Hence, the action space $\mathcal{A}$ of the reinforcement learning problem is given by $\mathcal{A} = \{drop, keep\}$.

To learn policies that generalize well to different environments, we describe the state of the robot and the environment in terms of general state features. We use the remaining battery life time in percent, the number of artificial landmarks left on board, and the distance to the artificial landmark that has been deployed closest to the robot. In addition to that, we make use of a feature that captures the abstract spatial structure of the environment based on a classification of the current position of the robot in terms of the categories room, doorway, corridor, and junction. There exist several robust techniques to compute this spatial feature for robots equipped with laser scanners [5], [11] or vision systems [10]. To efficiently represent $\hat{Q}^{\pi}$, we divide the state-action pairs into bins.

D. Statistical Convergence Test

As mentioned above, in actor-critic reinforcement learning, the critic becomes the actor after having observed enough episodes to learn the Q-function under the policy π followed by the current actor. To test whether the critic is already confident of the estimated Q-function, our approach applies a statistical convergence test after each episode.

Since the policy π observed by one critic is not changed in between the episodes, the estimated Q-function is computed using independent and identically distributed samples from the same policy. Therefore, given the definition in Eq. (5) of the estimator $\hat{Q}^{\pi}(s, a)$, its variance can be estimated as

$$S^2 = \frac{\sum_e \left(R^e_{\text{first}}(s, a) - \hat{Q}^{\pi}(s, a) \right)^2}{n_{\mathcal{F}} - 1}. \tag{11}$$

With confidence $1 - \alpha$, the value $Q^{\pi}(s, a)$ that we estimate lies in the interval

$$\left[\hat{Q}^{\pi}(s, a) - \sqrt{\frac{S^2}{n_{\mathcal{F}}}} t_{n_{\mathcal{F}}-1, \frac{\alpha}{2}}, \hat{Q}^{\pi}(s, a) + \sqrt{\frac{S^2}{n_{\mathcal{F}}}} t_{n_{\mathcal{F}}-1, \frac{\alpha}{2}} \right],$$

where $t_{n_{\mathcal{F}}-1, \frac{\alpha}{2}}$ is the $(1 - \frac{\alpha}{2})$-quantile of the Student's t-distribution with $n_{\mathcal{F}} - 1$ degrees of freedom. Once the confidence interval of the critic's estimate indicates convergence for all observed state-action pairs (s, a), the critic becomes actor, and a new critic is initialized.

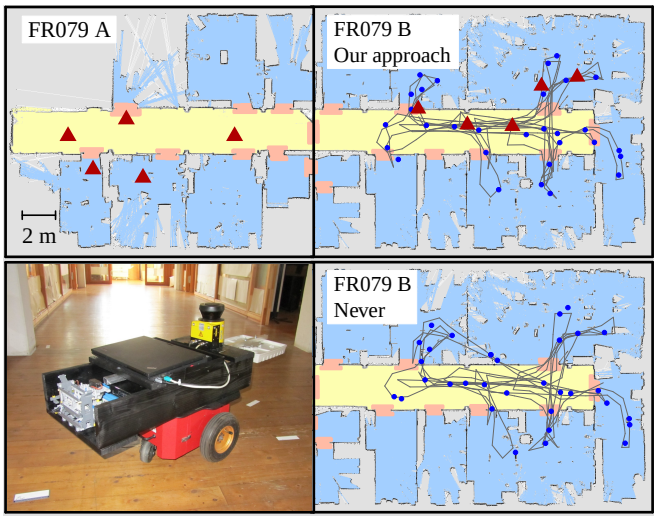

Fig. 2. The environment used for the experiments with the real robot and for cross-validation. Upper left: Deployed landmarks (red triangles) of one sample execution of our policy in the cross-validation experiment. Upper right: One of the runs of the real robot. Depicted are the estimated path (gray lines), the estimated positions of the environment features (blue dots), and the estimated positions of the deployed landmarks (red triangles). Lower right: Estimation for the same run without integrating the observations of the deployed landmarks. The environment is labeled with the spatial features used for learning: corridors (yellow), doorways (orange) and rooms (blue). The picture shows the Pioneer P3-DX robot used in the experiments.

V. Experimental Evaluation

We evaluated the performance of our approach both in simulation and on a real robot. In the experiments, we considered a robot that is equipped with a device for deploying five artificial landmarks, a noisy odometer, and a noisy landmark detection sensor. In the learning phase, we initialized the robot in each episode at a random pose and let it perform randomly sampled navigation tasks until its battery was empty. During operation, the robot applied a graph-based approach to SLAM using the framework proposed in [9] and a nearest neighbor filter to compute the data associations.

A. Experiments with a Real Robot

To evaluate the performance of our approach in practice, we applied a policy learned by our approach on the robot depicted in Fig. 2 executing randomly sampled paths in the environment shown on the right hand side of the figure. The learning phase was done in simulation, as it required 2,800 episodes to converge, which took 37.48 minutes in our multi-threaded implementation on an Intel® Core™ i7 2.8GHz.

The robot is equipped with a SICK RFI641 RFID reader with a circular field of view with radius 0.9 m mounted at the front and a custom made device for dropping RFID tags mounted in the back. Additionally, the robot is equipped with a SICK S300 laser range finder with a field of view of 270°, which we used for computing the spatial features. To do so, we applied a straightforward heuristic: it considers local minima in the scans for extracting door posts and long

TABLE I

EVALUATION OF REAL-WORLD EXPERIMENTS

	Error E	Translational Error	Rotational Error
Our approach	19.30	0.88 m	0.09 rad
Never	23.10	3.68 m	0.24 rad

parallel lines for finding corridor walls. Note that applying a more sophisticated classification technique [5], [11], would possibly even further improve our results. As environment features, we placed 70 RFID tags at randomly selected positions. The uniquely identifiable IDs of these tags, which the robot's SLAM system did not use, make it possible to precisely evaluate the data association error E introduced in Sec. IV-A. Furthermore, we evaluated the accuracy of the resulting pose estimates according to the framework described in Burgard *et al.* [3], using laser-based Monte Carlo localization to obtain reference positions.

Table I shows the results averaged over ten runs of the robot. It compares the performance of the estimation of the SLAM graph considering the deployed markers (*Our approach*) against considering only the environment features (*Never*). As can be seen in the table, the moderate reduction of the error E results in a large improvement of the pose estimation errors. The results of this experiment show that our approach is applicable in practice and that for the very noisy distance readings of the RFID sensor, the landmarks deployed by our approach especially help reducing the pose estimation errors.

B. Data Association Using the Learned Policies

In the first set of simulation experiments, we evaluated the data association performance of the policies learned by our approach. In this and the following experiments, we simulated the robot's landmark detection sensor with a circular field of view with radius 2 m and applied a deterministic spatial feature detection. We compared our learned policies to four naive approaches, namely *Equidistant*, which deploys the artificial landmarks equidistantly in time, *Random*, which deploys the markers at random time steps, *Always*, which deploys landmarks at every time step until all markers are deployed, and *Never*, which never deploys any landmarks, and to a heuristic in the sense of Kleiner *et al.* [8], named *Density*. This heuristic computes the obstacle density to the left and to the right of the robot from a simulated laser scan by applying kernel density estimation. Likewise, it computes the density of the already deployed landmarks. Based on these densities, it decides whether to drop a landmark. We used scenario-specific hand tuned parameters to optimize the performance of this heuristic.

We evaluated every approach in 100 randomly sampled simulated runs in a 6×6-row Manhattan-like environment with 96 environment features, for which a sample run can be seen in Fig. 1. In this environment, the simulated robot is able to discern the spatial features "corridor" and "junction". Fig. 3 shows the errors for the evaluated approaches. We additionally performed two-sided t-tests, which showed that

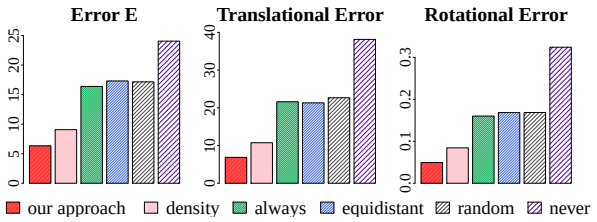

Fig. 3. The incorrectly estimated feature correspondences E and the translational and rotational pose estimation errors (in m and rad, respectively), averaged over 100 sample runs in a Manhattan-like environment.

TABLE II

CROSS-VALIDATION OF THE ERROR E IN FIVE ENVIRONMENTS

	Intel A	Intel B	FR079 A	FR079 B	FR106	Average
Our approach	4.30	0.45	2.62	2.39	5.29	3.01
Density	4.95	0.53	3.03	2.74	7.99	3.85
Always	6.18	1.36	5.75	4.87	10.30	5.69
Equidistant	7.09	1.80	7.12	5.84	10.59	6.49
Random	6.65	4.93	7.68	7.34	13.07	7.93
Never	14.32	10.24	15.58	14.64	26.67	16.29

our approach significantly outperforms all other approaches in all errors on a 95% confidence level.

C. Generalization to New Environments

We performed a five-fold cross validation in simulation to evaluate how well the policies computed by our approach generalize to environments that the robot has not seen previously. To do so, we considered five environments: FR079 A and FR079 B, depicted in Fig. 2, Intel A and Intel B, which capture parts of the well-known Intel Research Lab map, and FR 106, an office building at Freiburg Campus. In the simulation, we placed environment features at randomly sampled locations in the maps. In each fold of the cross validation, we learned a policy in four of the five environments and then evaluated its performance on the excluded one.

The resulting E values are given in the first row of Table II. In the other rows of the table, the E values for the heuristics described in Sec. V-B are stated for comparison. As can be seen in the table, the policies learned by our approach yield the lowest error values on average and for every single environment. This suggests that the policies computed by our approach generalize well to new environments.

D. Adaptation to the Sensor Range

In this section, we evaluate how the policies computed by our learning approach adapt to the range of the landmark detector. We learned landmark deployment policies for three simulated robots with the sensor ranges 2 m, 1 m, and 0.5 m in the FR106 environment also used in the previous experiments. Fig. 4 presents intensity plots of the resulting policies. As can be seen in the figure, the robot with sensor range 0.5 m strongly prefers deploying landmarks in doorways, because landmarks deployed in narrow passages are more likely to be observed later on, even with the small sensor range. The figure also shows that with increasing sensor range, the decision to deploy a landmark is stronger influenced by the distance to the nearest landmark and less

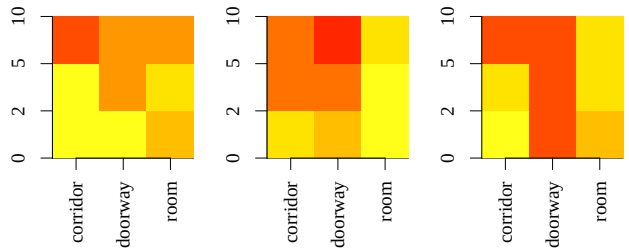

Fig. 4. The policies learned by our approach when using sensor ranges of 2 m (left), 1 m (middle), and 0.5 m (right), where red corresponds to $p(drop \mid s) = 1$ and white corresponds to $p(drop \mid s) = 0$. The ordinate is the distance to the nearest deployed landmark and the abscissa is the spatial feature. The values are averaged over the battery level and the number of remaining landmarks. The probability in the lower right corner cell is not converged due to the seldom occurrences of this situation.

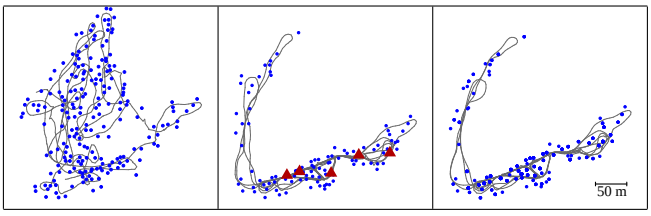

Fig. 5. Estimated landmark positions and robot path for the Victoria Park data set with data associations calculated without deployed landmarks (left), by taking into account the simulated observations of the landmarks deployed by our approach (middle), and for the ground truth data associations (right).

by the spatial feature. The results suggest that the policies computed by our learning approach adapt well to the range of the robot's sensor.

E. Large Scale Outdoor Environment

In this simulation experiment, we apply our approach on the well-known Victoria Park data set, covering an outdoor area of more than 200×200 m and including the observations of tree trunks. In this scenario, no spatial features are available, so our policy uses only the other three dimensions of the state space for selecting when to deploy a landmark. Note that due to the nature of the data, the policy was learned and evaluated on the same run, adding simulated observations of the deployed landmarks to the data set. Fig. 5 shows the estimated path of the robot with and without integrating the simulated observations of the deployed landmarks, as well as for applying the publicly available ground truth data associations (as used in [9]) for this data set. In this scenario, the vanilla nearest neighbor filter performs especially bad when not considering the deployed landmarks. Using a more sophisticated data association approach would certainly increase the performance here. In this scenario, our approach was able to reduce the number of incorrectly estimated feature correspondences E to 10, while *Equidistant* deployment resulted in a value of 20, *Always* in a value of 26, *Random* in a value of 78, and *Never* in a value of 130. The *Density* heuristic was not applicable, as no obstacle density can be calculated in this scenario. The results suggest that our approach works well even in large scale environments without spatial features.

VI. Conclusions

In this paper, we presented an approach based on actor-critic Monte Carlo reinforcement learning to learn a policy that allows a mobile robot to effectively deploy uniquely identifiable artificial landmarks so as to minimize data association errors in SLAM. Our approach uses features that support transferring the learned policies to previously not observed environments. Extensive experiments, both in simulation and on a real robot, demonstrate that our deployment approach results in significantly more accurate pose estimates than those obtained with different heuristics. In future work, we plan to utilize the spatial features defined in this paper not only for the deployment policy but also directly for data association. This could be quite helpful as observations made at positions with the same spatial feature are more likely to correspond to the same landmark than observations made at locations with different spatial features.

References

[1] M.A. Batalin and G.S. Sukhatme. Efficient exploration without localization. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2003.

[2] M.A. Bender, A. Fernãdez, D. Ron, A. Sahai, and S. Vadhan. The power of a pebble: Exploring and mapping directed graphs. *Information and Computation*, 2002.

[3] W. Burgard, C. Stachniss, G. Grisetti, B. Steder, R. Kümmerle, C. Dornhege, M. Ruhnke, A. Kleiner, and J.D. Tardós. A comparison of SLAM algorithms based on a graph of relations. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2009.

[4] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. Map validation and robot self-location in a graph-like world. *Robotics and Autonomous Systems*, 1997.

[5] S. Friedman, H. Pasula, and D. Fox. Voronoi random fields: Extracting topological structure of indoor environments via place labeling. In *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, 2007.

[6] G. Grisetti, D. L. Rizzini, C. Stachniss, E. Olson, and W. Burgard. Online constraint network optimization for efficient maximum likelihood map learning. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2008.

[7] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 2008.

[8] A. Kleiner, J. Prediger, and B. Nebel. RFID technology-based exploration and SLAM for search and rescue. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2006.

[9] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[10] J. Luo, A. Pronobis, B. Caputo, and P. Jensfelt. Incremental learning for place recognition in dynamic environments. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2007.

[11] O. Martinez-Mozos, R. Triebel, P. Jensfelt, A. Rottmann, and W. Burgard. Supervised semantic labeling of places using information extracted from sensor data. *Robotics and Autonomous Systems*, 2007.

[12] J. Neira and J.D. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 2001.

[13] E. Olson. Recognizing places using spectrally clustered local matches. *Robotics and Autonomous Systems*, 2009.

[14] H. Strasdat, C. Stachniss, and W. Burgard. Which landmark is useful? Learning selection policies for navigation in unknown environments. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2009.

[15] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 1998.

[16] S. Thrun. Finding landmarks for mobile robot navigation. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 1998.

[17] H. Wang, M. Jenkin, and P. Dymond. The relative power of immovable markers in topological mapping. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2011.