Improved Simultaneous Localization and Mapping using a Dual Representation of the Environment

Kai M. Wurm Cyrill Stachniss Giorgio Grisetti Wolfram Burgard

University of Freiburg, Dept. of Computer Science, Georges-Köhler-Allee 79, 79110 Freiburg, Germany

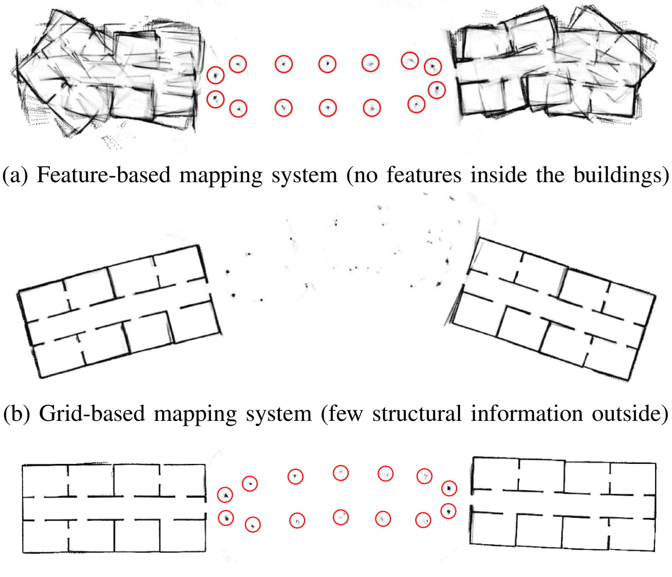
***Abstract*— The designer of a mapping system for mobile robots has to choose how to model the environment of the robot. Popular models are feature maps and grid maps. Depending on the structure of the environment, each representation has certain advantages. In this paper, we present an approach that maintains feature maps as well as grid maps of the environment. This allows a robot to update its pose and map estimate based on the representation that models the surrounding of the robot in the best way. The model selection procedure is obtained by reinforcement learning and takes a decision based on the current observation. As we will illustrate in simulation as well as in real world experiments, this allows a robot to learn accurate maps in a more robust way than approaches using only feature or only grid maps.**

I. Introduction

Building maps is one of the fundamental tasks of mobile robots. In the literature, the mobile robot mapping problem is often referred to as the *simultaneous localization and mapping (SLAM)* problem. It is considered to be a complex problem, because for localization a robot needs a consistent map and for acquiring a map a robot requires a good estimate of its location. This mutual dependency between the pose and the map estimates makes the SLAM problem hard and requires searching for a solution in a high-dimensional space.

A large variety of different estimation techniques has been proposed. Extended Kalman filter, sparse extended information filters, maximum likelihood methods, particle filter, and several other techniques have been applied to estimate the pose of the robot as well as a map of the environment. Most approaches to mapping use sets of features to model the environment, grid maps, or topological maps. Each representation has its own advantages. The environment the robot is deployed in mainly influences the decision which model to chose. For example, in large open spaces with predefined landmarks, feature-based approaches are likely to outperform mapping techniques based on grid maps. In dense and cluttered environment, however, grids offer substantial advantages.

In our system, we maintain the joint posterior about the trajectory of the robot and the map of the environment using a Rao-Blackwellized particle filter. The contribution of this paper is a novel approach which combines feature-based models with occupancy grid maps. Our approach allows a robot to perform its corrections based on both representations. It selects the model that is currently the best one to map the surroundings of the robot. The model selection process is obtained using reinforcement learning. It makes a decision based on the current sensor observations and the state of the filter. As we will demonstrate in the experiments, our approach outperforms pure grid and pure feature-based approaches. A motivating example is shown in Figure 1.


(a) Feature-based mapping system (no features inside the buildings)

(b) Grid-based mapping system (few structural information outside)

(c) Combining features and grid maps

Fig. 1. When mapping environments that contain large open spaces with few landmarks as well as dense structures, a combination of feature maps and grids maps outperforms the individual techniques.

This paper is organized as follows. After a discussion of related work, we briefly introduce mapping with Rao-Blackwellized particle filters in Section III. Section IV presents our filter for mapping which maintains the dual model of the environment. Section V explains our model selection process based on reinforcement learning. Experiments carried out in simulation and on real robots are presented in Section VI.

II. Related Work

Mapping techniques for mobile robots can be roughly classified according to the map representation and the underlying estimation technique. One popular map representation is the occupancy grid. Whereas such grid-based approaches are computationally expensive and typically require a huge amount of memory, they are able to represent arbitrary objects. Feature-based representations are attractive because of their compactness. However, they rely on predefined feature extractors, which assumes that some structures in the environments are known in advance.

The model of the environment and the applied state estimation technique are often coupled. One of the most popular approaches are extended Kalman filters (EKFs) in combination with landmarks. The effectiveness of the EKF approaches comes from the fact that they estimate a fully correlated posterior about landmark maps and robot poses [10, 15]. Their weakness lies in the strong assumptions that have to be made on both the robot motion model and the sensor noise. Moreover, the landmarks are assumed to be uniquely identifiable. There exist techniques [14] to deal with unknown data association in the SLAM context, however, if these assumptions are violated, the filter is likely to diverge [5, 9, 19].

Thrun *et al.* [18] proposed a method that uses the inverse of the covariance matrix. The advantage of the sparse extended information filters (SEIFs) is that they make use of the approximative sparsity of the information matrix and in this way can perform predictions and updates in constant time. Eustice *et al.* [4] presented a technique to make use of exactly sparse information matrices in a delayed-state framework.

In a work by Murphy, Doucet, and colleagues [2, 13], Rao-Blackwellized particle filters (RBPF) have been introduced as an effective means to solve the SLAM problem. Each particle in a RBPF represents a possible robot trajectory and a map. The framework has been subsequently extended by Montemerlo *et al.* [11, 12] for approaching the SLAM problem with landmark maps. To learn accurate grid maps, RBPFs have been used by Eliazar and Parr [3] and Hähnel *et al.* [7]. Whereas the first work describes an efficient map representation, the second presents an improved motion model that reduces the number of required particles. The work of Grisetti *et al.* [6] describes an improved variant of the algorithm proposed by Hähnel *et al.* [7] combined with the ideas of FastSLAM2 [11]. Instead of using a fixed proposal distribution, the algorithm computes an improved proposal distribution on a per-particle basis on the fly.

So far, there exist only very few methods that try to combine feature-based models with grid maps. One is the hybrid metric map (HYMM) approach [8]. It estimates the location of features and performs a triangulation between them. In this triangulation, a so called dense map is maintained which can be transformed according to the update of the corresponding landmarks. This allows the robot to obtain a dense map by using a feature-based mapping approach. However, it is still required that the robot is able to reliably extract landmarks.

III. Mapping with Rao-Blackwellized Particle Filters

According to Murphy [13], the key idea of the Rao-Blackwellized particle filter for SLAM is to estimate the joint posterior $p(x_{1:t}, m \mid z_{1:t}, u_{1:t-1})$ about the map m and the trajectory $x_{1:t} = x_1, \ldots, x_t$ of the robot. This estimation is performed given the observations $z_{1:t} = z_1, \ldots, z_t$ and the odometry measurements $u_{1:t-1} = u_1, \ldots, u_{t-1}$ obtained by the mobile robot. The Rao-Blackwellized particle filter for SLAM makes use of the following factorization

$$\begin{aligned} & p(x_{1:t}, m \mid z_{1:t}, u_{1:t-1}) = \\ & \qquad p(m \mid x_{1:t}, z_{1:t}) \cdot p(x_{1:t} \mid z_{1:t}, u_{1:t-1}). \end{aligned} \tag{1}$$

This factorization allows us to first estimate only the trajectory of the robot and then to compute the map given that trajectory. Since the map strongly depends on the pose estimate of the robot, this approach offers an efficient computation. This technique is often referred to as Rao-Blackwellization.

Typically, Eq. (1) can be calculated efficiently since the posterior about maps $p(m \mid x_{1:t}, z_{1:t})$ can be computed analytically using "mapping with known poses" since $x_{1:t}$ and $z_{1:t}$ are known.

To estimate the posterior $p(x_{1:t} \mid z_{1:t}, u_{1:t-1})$ about the potential trajectories, one can apply a particle filter. Each particle represents a potential trajectory of the robot. Furthermore, an individual map is associated with each sample. The maps are built from the observations and the trajectory represented by the corresponding particle.

This framework allows a robot to learn models of the environment and estimate its trajectory but it leaves open how the environment is represented. So far, this approach has been applied using feature-based models [11, 12] or grid maps [3, 6, 7, 13]. Each representation has its own advantages and one typically needs some prior information about the environment to select the appropriate model. In this paper, we combine both types of maps to represent the environment. This allows us to combine the advantages of both worlds. Depending on the most recent observation, the robot selects that model which is likely to be the best model in the current situation.

IV. Dual Model of the Environment

Our mapping system applies such a Rao-Blackwellized particle filter to maintain the joint posterior about the trajectory of the robot and the map of the environment. In contrast to previous algorithms, each particle carries a grid map as well as a map of features. The key idea is to maintain both representations simultaneously and to select in each step the model that is best suited to update the pose and map estimate of the robot. Our approach is independent of the actual features that are used. In our current system, we use a laser range finder and extract clusters of beam end points which are surrounded by free space. In this way, we obtain features from trees, street lamps, etc. Note that other feature detectors can be directly integrated into our approach. The detector itself is completely transparent to the algorithm.

In each step, our algorithm considers the current estimate as well as the current sensor and odometry observation to select either the grid or the feature model to perform the next update step. This decision affects the proposal distribution in the particle filter used for mapping. The proposal distribution is used to obtain the next generation of particles as well as to compute the importance weights of the samples.

In the remainder of this section, we first introduce the particularities of our particle filter. We then explain in the subsequent section how to actually select the model for the current step.

If the grid map is to be used, we draw the new particle poses from an improved proposal distribution as introduced by Grisetti *et al.* [6]. This proposal performs scan-matching on a per particle basis and then approximates the likelihood function by a Gaussian. This technique has been shown to yield accurate grid maps of the environment, given that there is enough structure to perform scan-matching for an initial estimate.

When using feature maps, we apply the proposal distribution as done by Montemerlo *et al.* [12] in the FastSLAM algorithm.

After the proposal is used to obtain the next generation of samples, the importance weights are computed according to Grisetti *et al.* [6] and Montemerlo *et al.* [12] respectively. Note that we compute for each sample i two weights $w_g^{(i)}$ (based on the grid map) and $w_f^{(i)}$ (based on the feature map). For resampling, one weight is required but we need both values in our decision process as explained in the remainder of this paper.

To carrying out the resampling step, we apply the adaptive resampling strategy originally proposed by Doucet [1]. It computes the so-called effective sample size or effective number of particles (N_{eff}) to decide whether to resample or not. This is done based on the weights resulting from the proposal used to obtain this generation of samples.

V. Model Selection

The probably most important aspect of our proposed algorithm is to decide which representation to choose given the current sensor readings and the filter. In the following, we describe different strategies we investigated and which are evaluated in the experimental section of this paper.

A. *Observation Likelihood Criterion*

A mapping approach that relies on scan-matching is most likely to fail if laser readings cannot be aligned to the map generated so far. This is likely to be the case in large open space with sparse observations. In such a situation it is often better to use a pre-defined feature extractor (in case there are feature) to estimate the pose of the robot.

A measure that can be used to detect such a situation is the likelihood $l(z_t, x_t, m_{g,t})$ that the scan-matching seeks to maximize. To point-wise evaluate the observation likelihood of a laser observation, we use the so called "beam endpoint model" [17]. In this model, the individual beams within a scan are considered to be independent. The likelihood of a beam is computed based on the distance between the endpoint of the beam and the closest obstacle from that point.

Calculating the average likelihood for all particles results in a value that can be used as a heuristic to decide which map representation to use in a given situation:

$$\bar{l} \;=\; \frac{1}{N}\sum_i l(z_t, x_t^{(i)}, m_{g,t}^{(i)}) \tag{2}$$

A heuristic for selecting the feature-based representation instead of the grid map can be obtained based on a threshold ($\bar{l} \leq c_1$).

B. *N_{eff} Criterion*

As described above, each particle i carries two weights $w_g^{(i)}$ and $w_f^{(i)}$, one for the grid-map and one for the feature-map. These weights can be seen as an indicator of how well a particle explains the data and therefore can be used as a heuristic for model selection. Since the weights of a particle are based on different types of measurement, they cannot be compared directly. What can be compared, however, is the weight distribution over the filter.

One way to measure this difference in the individual weights is to compute the variance of the weights. Intuitively a set of weights with low variance does not strongly favor any of the hypothesis represented by the particles, while a high variance indicates that some hypotheses are more likely than others.

This suggests that a strategy based on the N_{eff} value, which is strongly related to the variance of the weights, can be a good heuristic. N_{eff} is computed for both sets of weights as

$$N_{\mathit{eff}}^g = \frac{1}{\sum_{i=1}^N (w_g^{(i)})^2} \quad \text{and} \quad N_{\mathit{eff}}^f = \frac{1}{\sum_{i=1}^N (w_f^{(i)})^2}. \tag{3}$$

It can be easily seen, that a higher variance in the weights yields a lower N_{eff} value. Assuming that a set of particles with a higher variance in the weights is usually more discriminative, it seems reasonable to switch to the feature-based model whenever $N_{\mathit{eff}}^f < N_{\mathit{eff}}^g$.

C. *Reinforcement Learning for Model Selection*

Both approaches described above are clearly heuristics. In this section, we describe how to use reinforcement learning to combine the heuristics while avoiding their pitfalls. The basic idea of reinforcement learning is to find a mapping from states S to actions A which maximizes a numerical reward signal r (see [16] for an introduction). Such a mapping is called a policy and can be learned by interacting with the environment. Inspired by the human learning method of trial and error, this class of learning algorithms perform a series of actions and analyze the obtained reward.

There exist a number of algorithms for reinforcement learning that differ most notably in the knowledge available about the environment. If it can be modeled as an Markov decision process for example, technics such as policy iteration can be applied. If no model of the environment is available, Monte Carlo methods or Temporal-Difference Learning (TD learning) should be applied. For our approach, we use the Sarsa algorithm [16] which is a popular algorithm among the TD methods and does not require a model of the environment. It learns an action-value function $Q(s, a)$ which assigns a value to state-action pairs. Those values can then be used to generate a policy (e.g., choose the action that has the highest value in a given state).

To apply this method to our model selection problem, we have to define the states S, the actions A, and the reward $r : S \rightarrow R$. Defining the actions is straight forward as $A = \{a_g, a_f\}$, where a_g defines the use of the grid map and a_f the use of the feature map.

The state set has to be defined in a way that it represents all necessary information about the sensor input and the filter to make a decision. To achieve this, our state consists of the average scan matching likelihood $\bar{l}$, a boolean variable given by $N_{\mathit{eff}}^f < N_{\mathit{eff}}^g$, and a boolean variable if a known feature has currently been detected or not. This results in

$$S := \{\bar{l}\} \times \{1_{N_{\mathit{eff}}^f < N_{\mathit{eff}}^g}\} \times \{1_{\text{ feature detected}}\}. \tag{4}$$

The value of $\bar{l}$ is divided into (here seven) discrete intervals ($0.0-0.15$, $0.16-0.3$, $0.31-0.45$, $0.46-0.6$, $0.61-0.75$, $0.76-0.9$, $0.91-1.0$), resulting in $7\times 2\times 2 = 28$ states. It is important to keep the number of states small since learning the policy otherwise may require too many computation resources (even as a preprocessing step which needs to be executed only once).

The policy is learned on simulated data where the true robot pose x_t^* is available in every time step t. We use the weighted average deviation from the true pose to define our reward-function. To avoid a punishment that result from wrong decisions in the past (e.g. a wrong rotation), we only use the deviation accumulated since the last evaluation step $t-1$:

$$r(s_t) \quad = \quad r(s_{t-1}) - \sum_{i=1}^{N} w^{(i)} \, ||x^{(i)} - x_t^*|| \qquad (5)$$

Deviations from the simulated path result in negative rewards. As mentioned in the previous section, each particle stores two weights. For calculating the weighted average, we use $w_g^{(i)}$ if the last action taken was a_g and $w_f^{(i)}$ if a_f was taken.

The environment for learning consists of building-like structures with hallways and an outdoor part that models a set of trees. We recorded a simulated path and executed the learning algorithm for 1000 times. During learning, we us an ϵ-greedy policy. In state s, a greedy policy chooses the action a which has the highest value $Q(s,a)$. In contrast to this, an ϵ-greedy policy allows exploratory actions by choosing a random action with likelihood ϵ.

This technique results in a policy that tells the robot when to select the feature-based representation and when to choose the grid map. Note that our approach to learn a strategy for making decisions is independent of the actual feature detector used. One could even use this approach to choose among multiple feature detectors. The overall mapping algorithm is depicted in Algorithm 1.

VI. Experiments

Our approach has been evaluated using simulated and real robot data. Real world experiments have been conducted using an ActivMedia Pioneer 2-AT robot equipped with a SICK LMS laser range finder. For generating the simulated data, we used the Carnegie Mellon Robot Navigation Toolkit.

The experiments have been designed to verify that our mapping approach is able to reduce the error compared to the purely feature-based technique (FastSLAM [12]) and to the purely grid-based approach [6]. In case the environment suggests the use of one single model, the result is obviously the same as using the original approach.

A. Simulation Experiments

The simulated environment used to test our approach is shown in Figure 2. It shows two symmetric buildings connected by an alley spanning 70 m in total. We simulated a laser range finder with a maximum range of 4m which is less than the distance between the trees in the alley (5m). The motivating example in the introduction of this paper shows example results obtained with the different approaches. Figure 1 (a) is the result of the purely feature-based FastSLAM approach. Since no features are found inside the building structures, the robot cannot correct its trajectory inside the buildings. In contrast, the path through the alley is well corrected.

Algorithm 1 Our combined approach

```
Require:
  $\mathcal{S}_{t-1}$, the sample set of the previous time step
  $z_{l,t}$, the most recent laser scan
  $z_{f,t}$, the most recent feature measurement
  $u_{t-1}$, the most recent odometry measurement
Ensure:
  $\mathcal{S}_t$, the new sample set

  maptype = decide($\mathcal{S}_{t-1}, z_{l,t}, z_{f,t}, u_{t-1}$)

  $\mathcal{S}_t = \{\}$
  for all $s_{t-1}^{(i)} \in \mathcal{S}_{t-1}$ do
    $< x_{t-1}^{(i)}, w_{g,t-1}^{(i)}, w_{f,t-1}^{(i)} m_{g,t-1}^{(i)}, m_{f,t-1}^{(i)} >= s_{t-1}^{(i)}$

    // compute proposal
    if (maptype = grid) then
      $x_t^{(i)} \sim P(x_t \mid x_{t-1}^{(i)}, u_{t-1}, z_{l,t})$
    else
      $x_t^{(i)} \sim P(x_t \mid x_{t-1}^{(i)}, u_{t-1})$
    end if

    // update importance weights
    $w_{g,t}^{(i)} = \text{updateGridWeight}(w_{g,t-1}^{(i)}, m_{g,t-1}^{(i)}, z_{l,t})$
    $w_{f,t}^{(i)} = \text{updateFeatureWeight}(w_{f,t-1}^{(i)}, m_{f,t-1}^{(i)}, z_{f,t})$

    // update maps
    $m_{g,t}^{(i)} = \text{integrateScan}(m_{g,t-1}^{(i)}, x_t^{(i)}, z_{l,t})$
    $m_{f,t}^{(i)} = \text{integrateFeatures}(m_{f,t-1}^{(i)}, x_t^{(i)}, z_{f,t})$

    // update sample set
    $\mathcal{S}_t = \mathcal{S}_t \cup \{< x_t^{(i)}, w_{g,t}^{(i)}, w_{f,t}^{(i)}, m_{g,t}^{(i)}, m_{f,t}^{(i)} >\}$
  end for

  for $i = 0$ to $N$ do
    if (maptype = grid) then
      $w^{(i)} = w_g^{(i)}$
    else
      $w^{(i)} = w_f^{(i)}$
    end if
  end for

  $N_{\text{eff}} = \frac{1}{\sum_{i=1}^{N}(w^{(i)})^2}$
  if $N_{\text{eff}} < T$ then
    $\mathcal{S}_t = \text{resample}(\mathcal{S}_t, \{w^{(i)}\})$
  end if
```

The purely grid-based approach [6] is able to correctly map the buildings but introduces large errors in the alley (see Figure 1 (b)). Due to the limited range of the sensor, too few obstacles are observed and therefore no scan registration is possible and thus the grid-based approach fails to map the alley appropriately.

In contrast to this, our combined approach using the learned policy is able to correct the trajectory of the robot all the time by selecting the appropriate model. It uses the grid maps inside

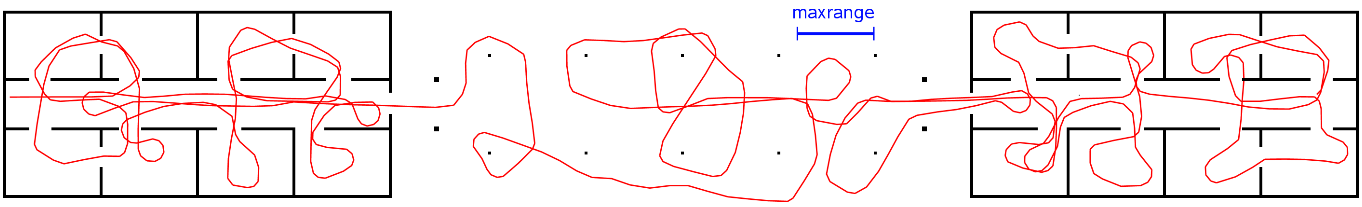


Fig. 2. Simulated environment used test our approach. This shows the ground truth map and trajectory of the robot.

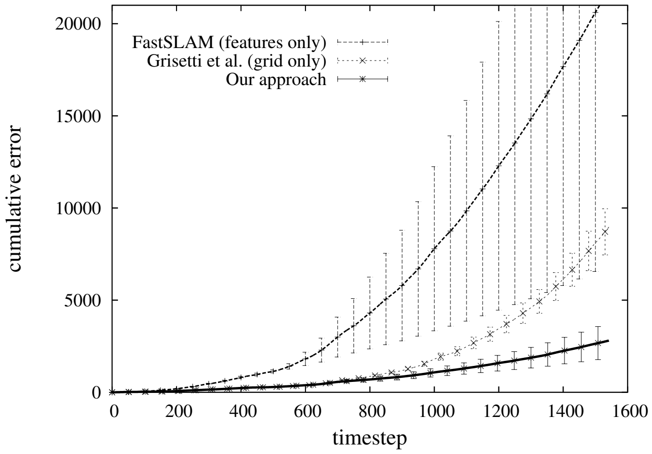


Fig. 3. Deviation of the weighted mean of the samples from ground truth using grid- and feature-model on their own and using the combined approach. The error bars illustrate the 0.05 confidence level.

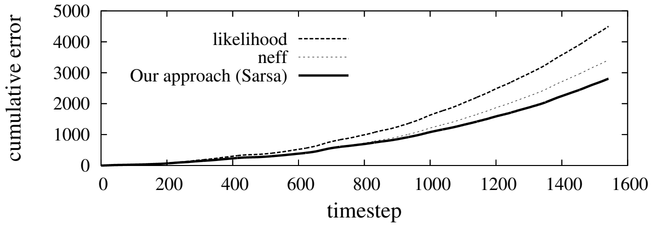


Fig. 4. Deviation of the weighted mean of the samples from ground truth using the scan-match likelihood heuristic, the N_{eff} heuristic and our approach.

the buildings and the features outside. The resulting map is shown in Figure 1 (c).

To evaluate our approach more quantitatively, we repeated this experiment for 20 times with different random seeds. We compared our approach to the pure feature-based approach and the pure grid-based approach. The results in Figure 3 show, that the combined approach is significantly better than both pure approaches (0.05 significance).

In addition, we compared the solution obtained by Sarsa with those of the scan-matching heuristic and the N_{eff} heuristic. We measured the absolute deviation from ground truth in every time step. Figure 4 illustrates that the average error of the learned model selection policy is lower than when using the heuristics. However, we could not show that this improvement is significant.

One interesting fact can be observed when comparing the results of these three technique by manual inspection. Even if the error measured as the deviation from the ground truth is not significantly smaller for the learned policy, the maps typically look nicer. The scan-match heuristic for example relies on a fixed threshold c_1. If the threshold is not optimally tuned, in can happen that the grid approach is not selected eventhough it would be better. This leads to walls which are more blurred or slightly missaligned. Figure 5 depicts a magnified view of two maps illustrating the difference. Unfortunately, it is hard to design a measure that is able to take this blurriness into account. A similar effect can be observed when using the N_{eff} criterion.

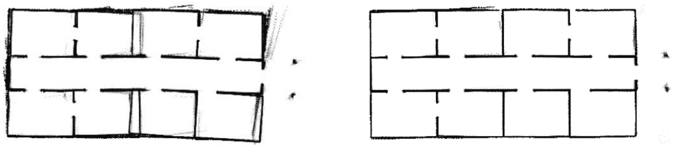


Fig. 5. Typical mapping results when using the likelihood-heuristic (left) and our Sarsa-based approach (right).

B. Real World Experiments

Real robot data has been recorded at Freiburg University. The computer science campus includes a parking space of about 50m by 120m (see Figure 6). Lamps are set in two rows at a distance of 16m. The dataset was recorded at a time when no cars were present and therefore only the lamps caused reflections of the laser beam. The robot was steered manually through a building, around the neighboring parking space, and back into the building. The trajectory is plotted in Figure 7. To evaluate our approach, we limited the maximum laser range to 12m, which is less than the distance between two lamps.

Since no ground truth was available, we measured the error to an approximated robot path which was generated using the grid-based approach of Grisetti *et al.* [6] with the full 80m sensor range (shown in red/dark gray in Figure 7). Due to the 80m range, the robot always observed enough obstacles to build an accurate map. Figure 8 shows the error of the weighted mean trajectory over time. In summary, the real robot experiment leads to similar results as simulated experiments. The combined approach performed better compared to both traditional SLAM techniques with 12m sensor range.

The computational requirements of the presented approach are approximatively the sum of the individual techniques. On a standard PC, our implementation runs online.

VII. Conclusions

In this paper, we presented an improved approach to learning models of the environment with a Rao-Blackwellized particle filters. Our approach maintains feature maps as well

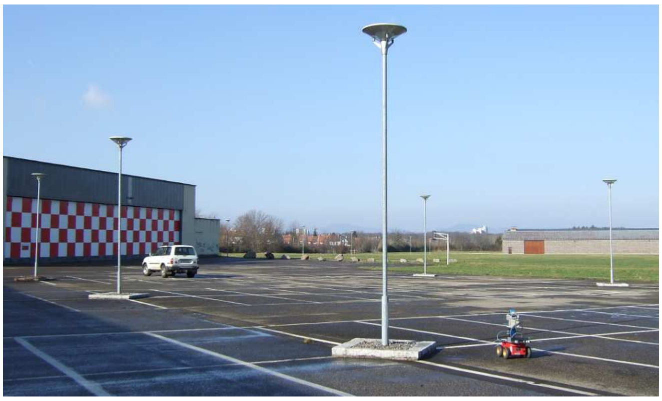


Fig. 6. Parking space at Freiburg campus.

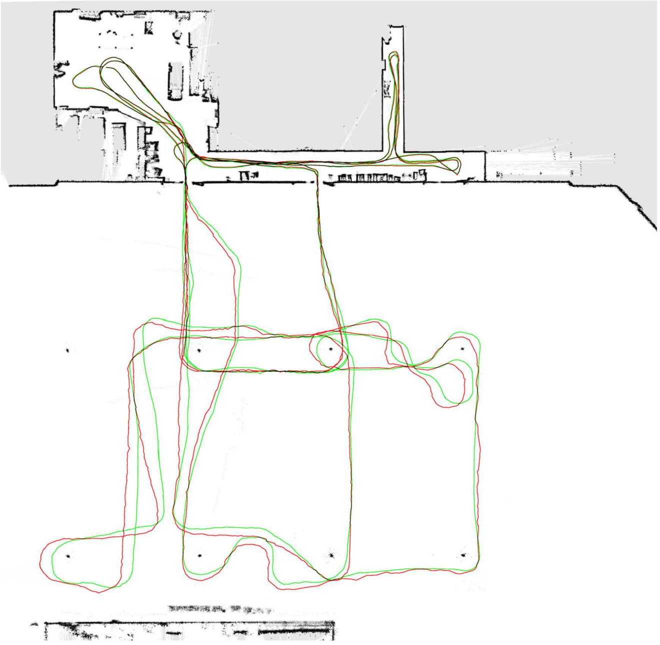


Fig. 7. Grid map of parking space and neighboring building 078 at Freiburg campus. The approximated robot trajectory is shown in red/dark gray, the result of our combined mapping approach is shown in green/light gray.

as grid maps to represent spatial structures. This allows the robot to select the model which provides the best expected map estimate. The model selection procedure is obtained by a reinforcement learning approach. The robot considers the previous estimate as well as the current observations to chose the model that will be used in the upcoming correction step. The process itself is independent of the actual feature detector. Our approach has been implemented and evaluated on real robot data as well as in simulation experiments. We showed that the presented technique allows a robot to more robustly learn maps of different types of environments. It ourperforms traditional approach that use only features or only grid maps.

Acknowledgment

This work has partly been supported by the German Research Foundation (DFG) under contract number SFB/TR-8 (A3), and by the EC under contract number FP6-2005-IST-6-RAWSEEDS, FP6-2005-IST-5-muFly, and FP6-004250-CoSy.

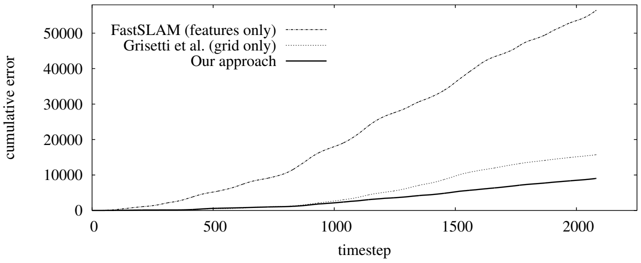


Fig. 8. Deviation of the weighted mean of the samples from the estimated trajectory (using the 80m range scanner).

References

[1] A. Doucet. On sequential simulation-based methods for bayesian filtering. Technical report, Signal Processing Group, Dept. of Engeneering, University of Cambridge, 1998.

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

[3] A. Eliazar and R. Parr. DP-SLAM: Fast, robust simultainous localization and mapping without predetermined landmarks. In *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, pages 1135–1142, Acapulco, Mexico, 2003.

[4] R. Eustice, H. Singh, and J.J. Leonard. Exactly sparse delayed-state filters. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2428–2435, Barcelona, Spain, 2005.

[5] U. Frese and G. Hirzinger. Simultaneous localization and mapping - a discussion. In *Proc. of the IJCAI Workshop on Reasoning with Uncertainty in Robotics*, pages 17–26, Seattle, WA, USA, 2001.

[6] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Transactions on Robotics*, 23(1):34–46, 2007.

[7] D. Hähnel, W. Burgard, D. Fox, and S. Thrun. An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 206–211, 2003.

[8] E.M. Nebot J.I. Nieto, J.E. Guivant. The hybrid metric maps (HYMMs): A novel map representation for denseslam. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2004.

[9] S. Julier, J. Uhlmann, and H. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proc. of the American Control Conference*, pages 1628–1632, Seattle, WA, USA, 1995.

[10] J.J. Leonard and H.F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*, 7(4):376–382, 1991.

[11] M. Montemerlo, S. Thrun D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, pages 1151–1156, 2003.

[12] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to simultaneous localization and mapping. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, pages 593–598, Edmonton, Canada, 2002.

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

[14] J. Neira and J.D. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897, 2001.

[15] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial realtionships in robotics. In I. Cox and G. Wilfong, editors, *Autonomous Robot Vehicles*, pages 167–193. Springer Verlag, 1990.

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

[17] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*, chapter Robot Perception, pages 171–172. MIT Press, 2005.

[18] S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani, and H. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *J. of Robotics Research*, 23(7/8):693–716, 2004.

[19] J. Uhlmann. *Dynamic Map Building and Localization: New Theoretical Foundations*. PhD thesis, University of Oxford, 1995.