Look-ahead Proposals for Robust Grid-based SLAM

Slawomir Grzonka, Christian Plagemann, Giorgio Grisetti, and Wolfram Burgard

University of Freiburg
Department of Computer Science
Georges-Koehler-Allee, 79110 Freiburg, Germany
{grzonka,plagem,grisetti,burgard}@informatik.uni-freiburg.de

Summary. Simultaneous Localization and Mapping (SLAM) is one of the classical problems in mobile robotics. The task is to build a map of the environment using on-board sensors while at the same time localizing the robot relative to this map. Rao-Blackwellized particle filters have emerged as a powerful technique for solving the SLAM problem in a wide variety of environments. It is a well-known fact for sampling-based approaches that the choice of the proposal distribution greatly influences the robustness and efficiency achievable by the algorithm. In this paper, we present a significantly improved proposal distribution for grid-based SLAM, which utilizes whole sequences of sensor measurements rather than only the most recent one. We have implemented our system on a real robot and evaluated its performance on standard data sets as well as in hard outdoor settings with few and ambiguous features. Our approach improves the localization accuracy and the map quality. At the same time, it substantially reduces the risk of mapping failures.

1 Introduction

The ability to construct models of natural and human-build environments is widely regarded as a precondition for autonomous service robots. Such models are required for basic tasks such as localization and motion planning. The combination of a Rao-Blackwellized particle filter (RBPF) with occupancy grid maps represents an effective and flexible solution to the SLAM problem as it only makes mild assumptions about the structure of the environment. Theoretically, given infinitely many particles, RBPFs always converge to the correct map. In practice however, only a finite number of particles can be used. This number inherently limits the uncertainty which can be represented by the filter and in this way can lead to divergence. Consider, for example, the environment depicted in the upper picture of Figure 1. This environment consists of two box-like landmarks, which cannot be perceived at the same time due to a limited sensor range. The geometric shape of these boxes needs to be represented accurately in the map as otherwise the estimate about the orientation of the robot relative to the landmark quickly gets lost. When the robot drives around a box several times the standard RBPF mapping approach typically turns the squared shape into a circle, since slight pose errors are introduced by the limited number of particles and by the suboptimal proposal distribution used. As a consequence,

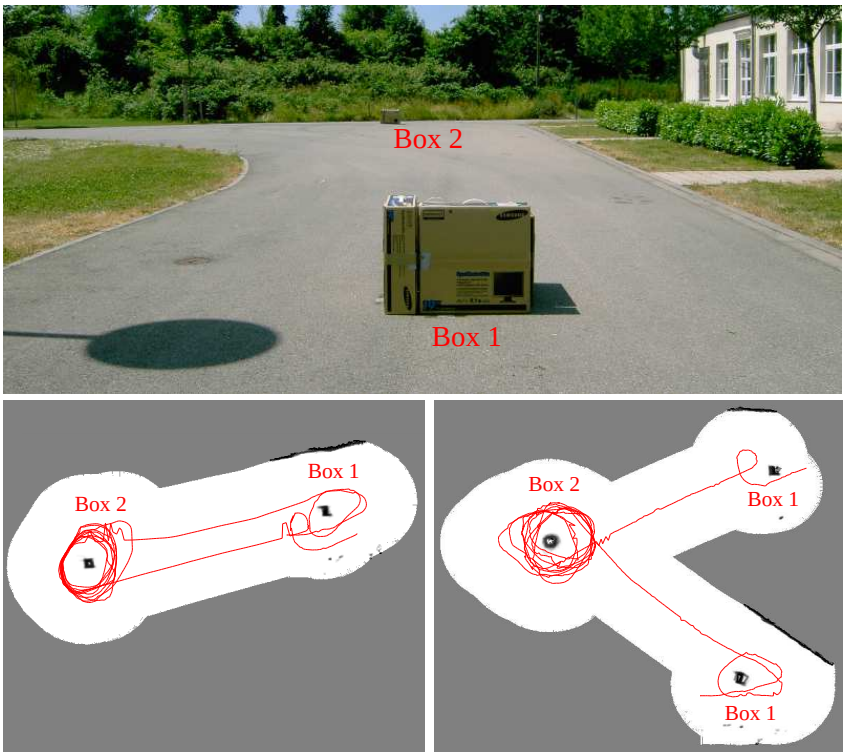

Fig. 1. A mobile robot with a limited sensor range navigates through the low structured environment depicted in the upper image, which consists of two cardboard boxes placed on flat, open terrain. The robot starts at *Box 1*, moves to *Box 2*, and revolves around it several times before returning to its origin. Standard mapping approaches cannot deal with the large amount of pose uncertainty build up when circling around the second box, which results in a seriously diverged map (lower right). In comparison, our approach using k-step look-ahead proposals retains the squared shape of the boxes and yields an accurate map (lower left). Note, that due to the limited laser range the hedge was visible only at Box 1.

the particle filter loses track of the heading of the robot and yields seriously wrong maps like the one depicted in the lower right diagram of Figure 1.

In this paper, we present the so-called look-ahead mapping approach which uses improved proposals for the robot pose to increase the accuracy of the particles after the proposal step. This is achieved by performing independent, k-step localization runs for the individual particles to better align them to their local maps. In this way, the filter can deal with higher levels of noise and operate in less structured environments. Experiments with real robots demonstrate that our approach can handle situations in which state-of-the-art RBPF-based approaches, like the one using a motion model based on the odometry error [10] or laser scan-matching error [5] or the one using dynamically adapted proposal distributions based on local laser scan-matching [4] fail.

2 Related Work

Originally, Murphy *et al.* [1, 10] have introduced Rao-Blackwellized particle filters as an effective means to solve the SLAM problem. In a RBPF, each particle represents a possible robot trajectory and a map. The framework has been subsequently extended by Montemerlo *et al.* [8, 9] for SLAM with discrete sets of landmarks. To efficiently learn accurate grid maps, Eliazar and Parr [3] described an efficient map representation. Additionally, Hähnel *et al.* [5] proposed an improved motion model that substantially reduces the number of particles required. Recently, Howard presented an extension towards multi-robot systems [6] in which he describes how to effectively merge the information obtained by different robots. Furthermore, Grisetti *et al.* [4] proposed an extension of the approach by Hähnel *et al.* [5]. Instead of using a fixed proposal distribution, their algorithm computes an improved proposal distribution on a per-particle basis, in a way similar to FastSLAM-2, as presented by Montemerlo *et al.* [7] for the case of landmark-based mapping.

All of the above mentioned approaches perform mapping and localization as the data is available, that is, the pose and the map uncertainty result from the very same set of observations processed. In the work presented here, we use k "future" measurements to build up a look-ahead proposal distribution for the robot pose. In this way, the uncertainty in the robot pose is reduced and consequently, less map uncertainty has to be represented by the filter.

3 Mapping with Rao-Blackwellized Particle Filters

The key idea of the Rao-Blackwellized particle filter for SLAM is to reason about possible robot trajectories and the corresponding maps using a sample-based representation [5]. Formally, the task is to estimate the joint posterior $p(\mathbf{x}_{1:t}, \mathbf{m} \mid \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$ of the map $\mathbf{m}$ and the trajectory $\mathbf{x}_{1:t} = \mathbf{x}_1, \ldots, \mathbf{x}_t$ of the robot, given observations $\mathbf{z}_{1:t} = \mathbf{z}_1, \ldots, \mathbf{z}_t$ and odometry measurements $\mathbf{u}_{1:t} = \mathbf{u}_1, \ldots, \mathbf{u}_t$. In the particle filter framework, the posterior after each time step is represented by a set of weighted trajectories $\mathbf{x}_{1:t}^{[j]}$ and the corresponding maps $\mathbf{m}_t^{[j]}$ generated from these trajectories. Using the factorization

$$p(\mathbf{x}_{1:t}, \mathbf{m}_t \mid \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) = p(\mathbf{m}_t \mid \mathbf{x}_{1:t}, \mathbf{z}_{1:t}) \cdot p(\mathbf{x}_{1:t} \mid \mathbf{z}_{1:t}, \mathbf{u}_{1:t}) \;, \quad (1)$$

we can derive a recursive filter that in each iteration updates the trajectory samples $\mathbf{x}_{1:t}^{i}$ and then analytically builds the corresponding map $\mathbf{m}_t^i$. Each filter iteration consists of the following steps:

1. *Sampling*: The next generation of particles $\{\mathbf{x}_t^{[j]}\}$ is obtained from the generation $\{\mathbf{x}_{t-1}^{[j]}\}$ by sampling from a proposal distribution π. Often, a probabilistic odometry-based motion model is used for π.

2. *Importance Weighting*: Importance weights $w_t^{[j]}$ are assigned to the individual particles according to

$$w_t^{[j]} = \frac{p(\mathbf{x}_{1:t}^{[j]} \mid \mathbf{z}_{1:t}, \mathbf{u}_{1:t})}{\pi(\mathbf{x}_{1:t}^{[j]} \mid \mathbf{z}_{1:t}, \mathbf{u}_{1:t})} \propto \frac{p(\mathbf{z}_t \mid \mathbf{m}_{t-1}^{[j]}, \mathbf{x}_t^{[j]}) p(\mathbf{x}_t^{[j]} \mid \mathbf{x}_{t-1}^{[j]}, \mathbf{u}_t)}{\pi(\mathbf{x}_t \mid \mathbf{x}_{1:t-1}^{[j]}, \mathbf{z}_{1:t}, \mathbf{u}_{1:t})} \cdot w_{t-1}^{[j]} \,. \tag{2}$$

   The weights account for the fact that the proposal distribution π is in general not equal to the target distribution of successor states [2].
3. *Resampling*: Particles are drawn with replacement proportional to their importance weight.
4. *Map Estimation*: For each particle, the corresponding map estimate $p(\mathbf{m}^{[j]} \mid \mathbf{x}_{1:t}^{[j]}, \mathbf{z}_{1:t})$ is computed based on the trajectory $\mathbf{x}_{1:t}^{[j]}$ of that sample and the history of observations $\mathbf{z}_{1:t}$.

The robustness and efficiency of this procedure strongly depends on the proposal distribution π that is used to sample the new state hypotheses in the selection step. If the proposal distribution differs too much from the true posterior, there is a high risk of filter divergence. In the following section, we introduce a concrete proposal distribution π that utilizes a set of "future" sensor measurements to yield better pose estimates. The resulting weight update equation is straightforward to implement, while the new approach is more robust in standard and hard, poorly-structured environments.

4 Look-ahead Proposals

The standard RBPF mapping approach fails in situations, in which the particle distribution significantly differs from the true posterior. This can happen when the proposal distribution π provides a bad approximation to the true one, or when the environment does not provide enough structure to allow proper particle weighting. The latter situation is illustrated in the lower right diagram of Figure 1. Due to the limited structure of the environment, the robot is unable to localize itself properly and looses the fine structure of Box 2. Before returning to its starting location, the robot is clearly delocalized, such that Box 1 appears twice in the resulting map. Such a divergence can either be avoided by using an extremely large number of particles or by directing the given number of particles to more accurate locations. We follow the latter strategy by computing the pose prediction in each iteration based on the k next sensory inputs instead of just one. These k measurements are used to better localize each particle within its own map. Concretely, for each mapping particle at time $t-1$, we draw l localization particles and localize them k steps ahead within their map. The resulting pose posterior at time $t+k$ is then used to sample the successor pose of the mapping particle at time t. This process is visualized in Figure 2.

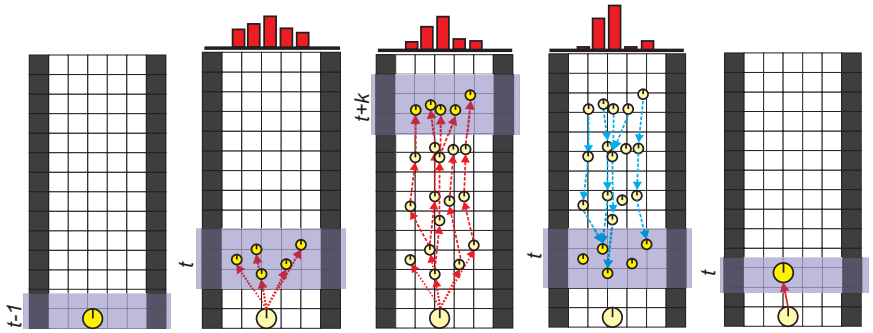

Fig. 2. The algorithm for look-ahead pose sampling using k steps: For each mapping particle at time $t-1$ (first diagram), draw l localizing particles (second). Move them according to u_{t+i} and weight them according to z_{t+i} until $t+k$ (third diagram). Propagate back the weights of the particles to the initial situation at time t (forth) and draw the successor state according to this distribution (right diagram). The charts above the diagrams visualize the current weights of the individual localization particles at the highlighted timeindex.

By using the additional sensory input, a more informed and thus more accurate proposal for the robot pose can be computed. Formally, the idea is to compute a better estimate for the pose $\mathbf{x}_t$, using the previous position $\mathbf{x}_{t-1}$, the commands (odometry) $\mathbf{u}_{t:t+k}$, and the measurements $\mathbf{z}_{t:t+k}$ up to time-index $t+k$. As stated in Equation (1), the Rao-Blackwellized particle filter is an approach for sequentially estimating the distribution

$$p(\mathbf{m}_t|\mathbf{x}_{1:t},\mathbf{z}_{1:t})\ p(\mathbf{x}_{1:t}|\mathbf{z}_{1:t},\mathbf{u}_{1:t})\ . \tag{3}$$

Here, $p(\mathbf{x}_{1:t}|\mathbf{z}_{1:t},\mathbf{u}_{1:t})$ is approximated using a particle filter, whose samples we will call *SLAM particles*. In the standard approach, the SLAM particles are drawn from a proposal distribution based on the motion model $p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{u}_t)$. In our approach, we use the more informed proposal $p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{u}_t,\mathbf{z}_{t:t+k},\mathbf{u}_{t+1:t+k},\mathbf{m}_{t-1})$. Such a proposal results from performing an independent k-step localization for each SLAM particle. The localization algorithm is initialized with the map and the robot pose of the corresponding SLAM particle at time $t-1$. Our proposal can be rewritten more compactly, omitting $\mathbf{m}_{t-1}$, as $p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{u}_{t:t+k},\mathbf{z}_{t:t+k})$, which we can rewrite as

$$p(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{u}_{t:t+k},\mathbf{z}_{t:t+k}) = \int p(\mathbf{x}_{t:t+k}|\mathbf{x}_{t-1},\mathbf{u}_{t:t+k},\mathbf{z}_{t:t+k})\ \mathrm{d}\mathbf{x}_{t+1:t+k}. \tag{4}$$

Given the poses $\mathbf{x}_t \ldots \mathbf{x}_{t+k}$ and the map $\mathbf{m}_{t-1}$ the obsersavtions $\mathbf{z}_t \ldots \mathbf{z}_{t+k}$ are independant. Therefore, we can rewrite the term inside the integral as

$$p(\mathbf{x}_{t:t+k}|\mathbf{x}_{t-1},\mathbf{u}_{t:t+k},\mathbf{z}_{t:t+k}) = \eta \prod_{j=t}^{t+k} p(\mathbf{z}_j|\mathbf{x}_j) \cdot \prod_{j=t}^{t+k} p(\mathbf{x}_j|\mathbf{x}_{j-1},\mathbf{u}_j). \tag{5}$$

Here $\eta = p(\mathbf{z}_{t:t+k}|\mathbf{x}_{t-1}\mathbf{u}_{t:t+k})^{-1}$ is the Bayes normalizer. A particle approximation of Equation 5 (the localizer particles) can be obtained by sampling a sequence of poses according to the sequence of motion commands. Each sample i has a weight $v^{(i)}_{t:t+k}$ proportional to the likelihood of the chain of observations starting at $t-1$.

$$\mathbf{x}^{(i)}_{t:t+k} \sim \prod_{j=t}^{t+k} p(\mathbf{x}^{(i)}_j|\mathbf{x}^{(i)}_{j-1}, \mathbf{u}_j) \qquad v^{(i)}_{t:t+k} = \prod_{j=t}^{t+k} p(\mathbf{z}^{(i)}_j|\mathbf{x}^{(i)}_j). \tag{6}$$

A sampled approximation of the integral in Eq. 4

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_{t:t+k}, \mathbf{z}_{t:t+k}) \sim \left\langle \mathbf{x}^{(i)}_t, v^{(i)}_t \right\rangle \tag{7}$$

is recovered from the samples in Eq. 6. This is done by truncating each trajectory $\mathbf{x}^{(i)}_{t:t+k}$ at time t and by back-propagating the weights of the trajectory, according to Fig. 2. Due to resampling operations, the evolution of these trajectories can be described as a tree [3], thus the sample $\mathbf{x}^{(i)}_t$ receives a weight $v^{(i)}_t$ which is the sum of the weights of its successors.

The set $\mathcal{S} = \left\{ \left\langle \mathbf{x}^{(i)}_t, v^{(i)}_t \right\rangle \right\}$ is a sampled approximation of our proposal distribution. We can draw from this set N new SLAM particles $\left\{ \mathbf{x}^{(j)}_t \right\}$ for time t, according to the importance weights. The true posterior $p(\mathbf{x}^{(j)}_t|\mathbf{x}^{(j)}_{t-1}, \mathbf{z}_t, \mathbf{u}_t)$ (the SLAM particles) is recovered from this set by assigning to each newly drawn sample $\mathbf{x}^{(j)}_t$ a weight $w^{[j]}_t$ according to the importance sampling principle (see Eq. 2):

$$w^{[j]}_t \propto w^{[j]}_{t-1} \frac{p(\mathbf{x}^{(j)}_t|\mathbf{x}^{(j)}_{t-1}, \mathbf{z}_t, \mathbf{u}_t)}{p(\mathbf{x}^{(j)}_t|\mathbf{x}^{(j)}_{t-1}, \mathbf{u}_{t:t+k}, \mathbf{z}_{t:t+k})} \tag{8}$$

$$\propto w^{[j]}_{t-1} \frac{p(\mathbf{z}_t|\mathbf{x}^{(j)}_t) p(\mathbf{x}^{(j)}_t|\mathbf{x}^{(j)}_{t-1}, \mathbf{u}_t)}{p(\mathbf{x}^{(j)}_t|\mathbf{x}^{(j)}_{t-1}, \mathbf{u}_{t:t+k}, \mathbf{z}_{t:t+k})} \tag{9}$$

$$\propto w^{[j]}_{t-1} \frac{p(\mathbf{z}_t|\mathbf{x}^{(j)}_t)}{v^{(j)}_t} \tag{10}$$

The last step follows directly from Equation 7 and the fact that the trajectories $\mathbf{x}^{(i)}_{t:t+k}$ had been drawn including the motion $\mathbf{u}_t$ between $t-1$ and t. Note that for updating the weight of a SLAM particle, we just need the weight of the drawn successor state at time t (first iteration of the nested localization run) and the corresponding weight at time $t+k(=v_t)$. Both values are readily available from the localization run. Equation (10) assures, that no information (odometry, measurements) is integrated more

than once. Thus, the presented approach does not change the estimated posterior distribution, but only the way it is represented by the limited particle set of the filter.

5 Experiments

In this section, we present a set of experiments demonstrating that our mapping approach can be applied in both indoor and outdoor scenarios. Furthermore, we compared our approach using localization proposals (termed *LP* in this section) to a state-of-the-art technique [4], which uses a dynamically adapted proposal based on scan matching (*SMP*) and to a standard RBPF using odometry-based proposals (*OP*). The results show that our approach is more robust in poorly structured situations, while also providing high quality maps in more structured environments.

Measuring Map Quality

To assess the quality of the resulting maps we evaluate a measure denoted as *revisiting accuracy* (RA), which reflects the error in the robot pose estimate at revisited places relative to previous visits. This measure captures the internal consistency of a map. It therefore better reflects the practical usefulness of the map than for example the absolute mean squared error of the corrected trajectory relative to an assumed true one. The latter measure is overly sensitive to distortions on a global level, which does in general not lead to practical problems, and it requires a ground truth trajectory.

For calculating the *revisiting accuracy*, we add color markers to the ground at places that are to be traversed several times during the experiment. We then recorded the timestamps at which the robot passed over these checkpoints. Let p be a checkpoint visited at times t_1 and t_2. The revisiting accuracy for this location is defined as

$$\epsilon^p_{t_1,t_2} = \sum_j w^{[j]} \sqrt{(1-\lambda)((x^{[j]}_{t_1} - x^{[j]}_{t_2})^2 + (y^{[j]}_{t_1} - y^{[j]}_{t_2})^2) + \lambda(\theta^{[j]}_{t_1} - \theta^{[j]}_{t_2})^2} \; .$$

Here, x, y, and θ are the components of the pose vector and $\lambda \in [0, 1]$ is a weighing factor for the rotational component. Intuitively, the revisiting accuracy for a given checkpoint is the (weighted) distance between the estimated positions within every map, weighted by the corresponding map-probability.

Mapping an Office Environment

We tested our approach in the highly structured office environment depicted in Figure 3. Given this dataset, we compared our approach (*LP*) to *SMP* and *OP* in terms of the revisiting accuracy measure defined above. Figure 3

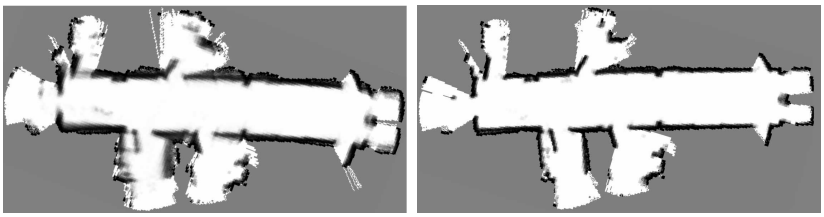

Fig. 3. Maps obtained from the indoor environment using *OP* (left) and *LP* (right).

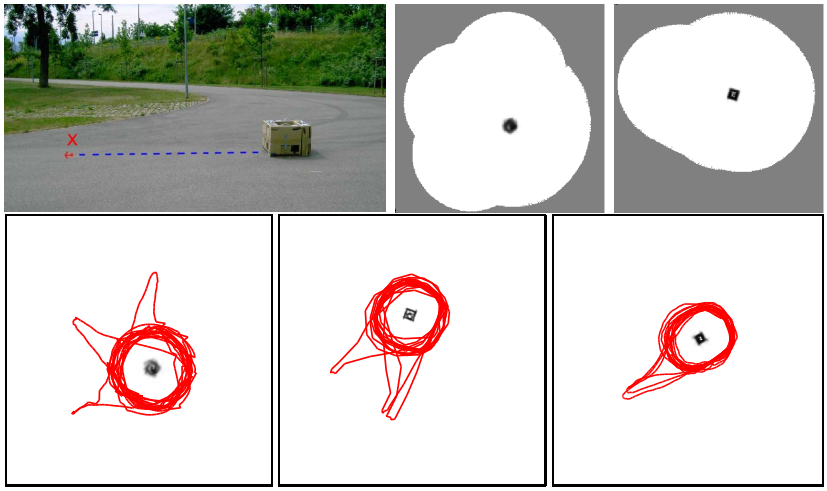

Fig. 4. Mapping a low-structured environment (upper left): Maps obtained by *OP* (upper middle) and our approach (upper right). Traveled paths of the robot using *OP* (lower left), *SMP* (lower middle) and our *LP* approach (lower right). All approaches used 50 mapping particles. Our approach used additionally 100 localization particles per map and a look-ahead of $k = 5$.

depicts the maps obtained using *OP* (left diagram) and our approach (right diagram) for 20 mapping particles. In our approach, we used 50 localization particles per map particle and a look-ahead of $k = 3$. The map generated using the *SMP* is similar to ours. The localization accuracy is between 5 cm and 10 cm for *OP* and less than 5 cm for both, our approach (*LP*) and *SMP*. Overall, a localization error of less than 10 cm is considerably small with the given grid resolution of 5 cm per cell.

Mapping a Low-Structured Outdoor Environment

We compared our approach to those proposed by Hähnel *et al.* [5] and Grisetti *et al.* [4] in the outdoor environment depicted in the upper left picture of Figure 4. The robot started at the checkpoint "X", moved around the cardboard box 15 times. Thereby it returned to the checkpoint every 5th run. The laser range was limited to 4 meters, so that the box was the only visible map element. Figure 4 depicts the mapping results of the *OP* and our

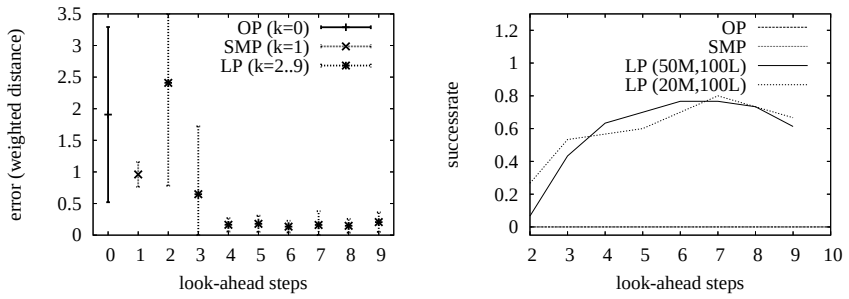

Fig. 5. Mean and standard deviation for the approaches (left) and the ratio of runs with an error less than 0.2 (right). Note that the *OP*-RBPF and the *SMP*-RBPF approach never had an revisiting accuracy error below 0.2. Thus the success rates of both approaches is 0.

LP approach (second and third upper diagram) using 50 SLAM particles. For generating the map, we used a look-ahead of $k = 5$ and 100 localization particles per SLAM particle. The three lower diagrams in Figure 4 depict typical paths estimated by the three mapping approaches. The distance between the checkpoint "X" and the box is around 4 m. As can be seen from these results, *OP* fails to map this environment and yields a highly inconsistent map. Although, when using *SMP*, the map quality is higher and the box has retained its squared shape, the resulting map is still inconsistent. Our approach, in contrast, is able to deal with this hard situation and generates accurate maps. To quantitatively evaluate the performance of the approaches, we analyzed the mapping results for different numbers of particles. Each approach was executed 30 times for each setting. We measured $\epsilon^X_{t_{\text{first}},t_{\text{last}}}$, with t_{first} and t_{last} being the timestamps, when the robot moved over the checkpoint "X" for the first and the last time respectively. Figure 5 (left) gives the results of this experiment. For reasons of better readability we depicted the results obtained by 50 mapping particles only. Furthermore, Figure 5 (right) depicts the success rate of the runs, defined as the ratio of runs where the revisiting error ϵ was less than 0.2.

It can be seen from these experiments, that in situations with hardly identifiable features, the performance obtained using our proposal is higher than the one obtained by the other grid based RBPF mappers [4, 5]. Although the RBPF based on *SMP* outperforms the standard RBPF, it achieves significantly less accurate maps than the technique proposed in this paper. Our approach is able to "look through gaps" of low structure until enough structure is provided for proper localization. In this way, it can handle such a hard environment even with less mapping particles (see figure 5 (right), curve obtained with 20 mapping particles and 100 localization particles for each map).

6 Conclusions

In this paper, we presented an extension to the Rao-Blackwellized particle filter for simultaneous localization and mapping (SLAM), which significantly improves the mapping quality and leads to accurate maps even in environments which are poorly structured. Our approach uses a novel proposal distribution that looks k steps "ahead in time" to yield a more informed pose estimate for the mapping decision. We provided a mathematical derivation of the approach and showed, that the weight update for our improved proposal takes a simple form and thus is easy to integrate into a standard RBPF framework. Our method has been implemented and tested on real robot data sets. We furthermore compared our technique to two state of the art mapping techniques. Experimental results suggest that our approach yields better maps, especially in environments with sparse features that facilitate accurate localization of particles.

References

1. 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.
2. A. Doucet, N. de Freitas, and N. Gordan, editors. *Sequential Monte-Carlo Methods in Practice*. Springer Verlag, 2001.
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. G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based slam with Rao-Blackwellized particle filters by adaptive proposals and selective resampling. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 2443–2448, Barcelona, Spain, 2005.
5. 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, Las Vegas, NV, USA, 2003.
6. Andrew Howard. Multi-robot simultaneous localization and mapping using particle filters. In *Robotics: Science and Systems*, Cambridge, MA, USA, 2005.
7. 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, Acapulco, Mexico, 2003.
8. M. Montemerlo and S. Thrun. Simultaneous localization and mapping with unknown data association using FastSLAM. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, Taipei, Taiwan, 2003.
9. 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.
10. 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.