

Look-ahead Proposals for Robust Grid-based SLAM with Rao-Blackwellized Particle Filters

Slawomir Grzonka Christian Plagemann
Giorgio Grisetti Wolfram Burgard

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

Abstract

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 robustness and efficiency achievable by the algorithm. In this paper, we present an improved proposal distribution for grid-based SLAM with Rao-Blackwellized particle filters, 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 and 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 truly autonomous service robots. Such models are required for a variety of fundamental tasks including 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 accuracy of the filter and too few particles can lead to a divergence of the filtering process.

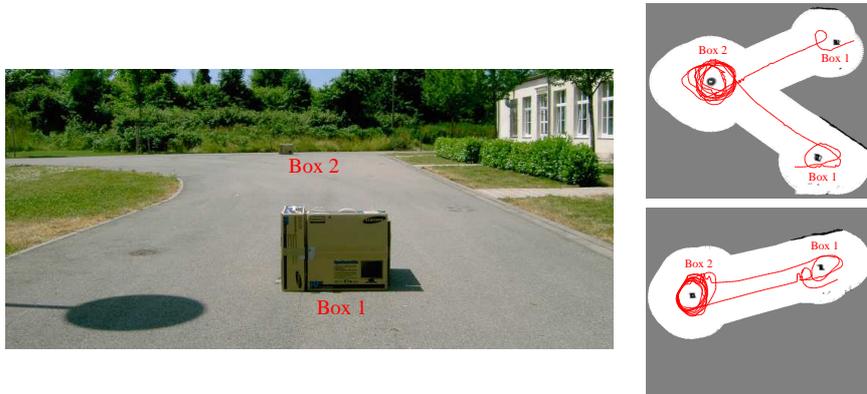


Figure 1: A mobile robot with a limited sensor range navigates through the low structured environment depicted in the left image, which consists of two cardboard boxes placed on flat, open terrain. The robot starts at *Box 1*, moves to *Box 2*, and revolves around it several times before returning to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

For example, consider the environment depicted in the left image of Figure 1. This environment consists of two box-like landmarks, which cannot be perceived at the same time due to the limited sensor range. When the robot moves around one of the boxes several times, the standard RBPF mapping approach with a limited number of particles typically is not able to accurately estimate the relative position of the robot to the box. As a result, the squared shape of the box is turned into a circle for all particles. The particle filter loses track of the heading of the robot and yields seriously wrong maps like the one depicted in the upper right diagram of Figure 1.

Some SLAM systems address situations in which there is too few structure in the environment by integrating multiple measurements into local maps. Those local maps are then used as input for a RBPF. In this paper, we propose an orthogonal approach which stays within the RBPF framework. The two techniques can be combined to obtain a further increase in performances. However, the major aim of this paper is to investigate the benefits of a k -step look-ahead algorithm which is able to compute more accurate proposal distributions.

Applied within the context of Rao-Blackwellized SLAM, our approach performs k -step localization runs for the individual particles to better align them to their 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 [Murphy, 1999] or laser scan-matching error [Hähnel *et al.*, 2003], or the one using dynamically adapted proposal distributions based on local laser scan-matching [Grisetti *et al.*, 2007] fail.

The article is organized as follows. After the discussion of related work in the next section, we give an introduction to the SLAM problem and discuss the Rao-Blackwellized particle filter. In Section 4, we introduce our k -step look-ahead proposals and discuss their integration into the general RBPF mapping framework. In Section 5, we give qualitative and quantitative results for real mapping scenarios.

2 Related Work

Particle filters have been applied to various kinds of robotic state estimation problems including localization [Fox *et al.*, 1999], mapping [Hähnel *et al.*, 2003, Eliazar and Parr, 2003, Grisetti *et al.*, 2007, Montemerlo *et al.*, 2003], or data association [Tipaldi *et al.*, 2007]. Murphy, Doucet, and colleagues were the first to present an approach based on a Rao-Blackwellized particle filter for learning grid maps [Doucet *et al.*, 2000, Murphy, 1999]. The first efficient approach for mapping with Rao-Blackwellized particle filters was the FastSLAM algorithm [Montemerlo *et al.*, 2002]. It uses a set of Kalman filters to represent the map features conditioned on a sampled robot trajectory. The particles are drawn from the odometry motion model and weighed by the likelihood of the observations. The grid-based variant [Hähnel *et al.*, 2003] performs scan-matching as a preprocessing step. In this way, they are able to draw samples from distributions with lower variances compared to proposals computed based on the odometry only. This reduces the number of required particles and allows a robot to estimate the map online. In contrast, [Eliazar and Parr, 2003] focus on an efficient grid map representation which allows the particles to share a map.

Howard presented an extension towards multi-robot systems [Howard, 2005] in which he describes how to effectively merge the information obtained by different robots. FastSLAM2 [Montemerlo *et al.*, 2003] has been subsequently proposed as an extension to the original FastSLAM algorithm. It computes an improved proposal based on the most recent sensor observation to restrict the space for sampling. A Gaussian approximation is computed for each particle by integrating the most recent odometry and the most recent landmark measurements via an extended Kalman filter. [Grisetti *et al.*, 2007] extended FastSLAM2 to deal with large-scale occupancy grid maps. This technique combines scan-matching on a per particle basis with informed proposal distributions. Recently, [Stachniss *et al.*, 2007] build on these ideas and propose to adaptively switch to more complex, i.e. non-Gaussian, proposal distributions when the posterior deviates too much from a Gaussian.

All of the above-mentioned approaches perform mapping and localization as the data is available. In this way, the pose and the map uncertainty are computed by processing for the very same set of observations. In contrast, this

work uses 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.

In parallel to our work, [Beevers and Huang, 2007] proposed to use look-ahead proposals for the landmark based case. They propose two strategies for incorporating the future information into the sampling process. The first approach consists in drawing samples from a so called “block proposal distribution”. The block proposal is obtained by computing the Gaussian over the last k states by means of an EKF. The second approach integrates the next k measurements at each step. The robustness of the filter is increased by smoothing the resulting trajectories. Conversely, we propose to *adapt* the sampling procedure in a particle filter-based framework by integrating the k last measurements in the spirit of [Pitt and Shephard, 1997]. We apply this general principle to the Rao-Blackwellized particle filter (RBPF) in the SLAM context. The basic idea is to delay the drawing of a successor state. This allows us to take into account observations acquired within a temporal interval and to improve the state prediction based on the odometry. In this way we reduce the risk of performing actions, which are locally beneficial but lead to filter degeneration at a later stage.

3 Mapping with Rao-Blackwellized Particle Filters

The key idea of the Rao-Blackwellized particle filter for SLAM is to track the possible robot trajectories and the corresponding maps using a sample-based representation [Hähnel *et al.*, 2003]. 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 $\mathbf{x}_{1:t} = \langle \mathbf{x}_1, \dots, \mathbf{x}_t \rangle$ of the robot, given observations $\mathbf{z}_{1:t} = \langle \mathbf{z}_1, \dots, \mathbf{z}_t \rangle$ and odometry measurements $\mathbf{u}_{1:t} = \langle \mathbf{u}_1, \dots, \mathbf{u}_t \rangle$. In the particle filter framework, the posterior after each time step t 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 $\mathbf{x}_{1:t}^{[j]}$ and then analytically builds the corresponding maps $\mathbf{m}_t^{[j]}$. Concretely, 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

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

The weights account for the fact that the proposal distribution π is in general not equal to the target distribution of successor states [Doucet *et al.*, 2001].

3. *Resampling*: Particles are drawn with replacement proportional to their importance weight.
4. *Map Estimation*: For each particle, the corresponding map estimate $p(\mathbf{m}_t^{[j]} | \mathbf{x}_{1:t}^{[j]}, \mathbf{z}_{1:t})$ 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 of 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 loses the fine structure of Box 2. Before returning to its starting location, the robot is clearly de-localized, such that Box 1 appears twice in the resulting map. Such an outcome 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 M 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.

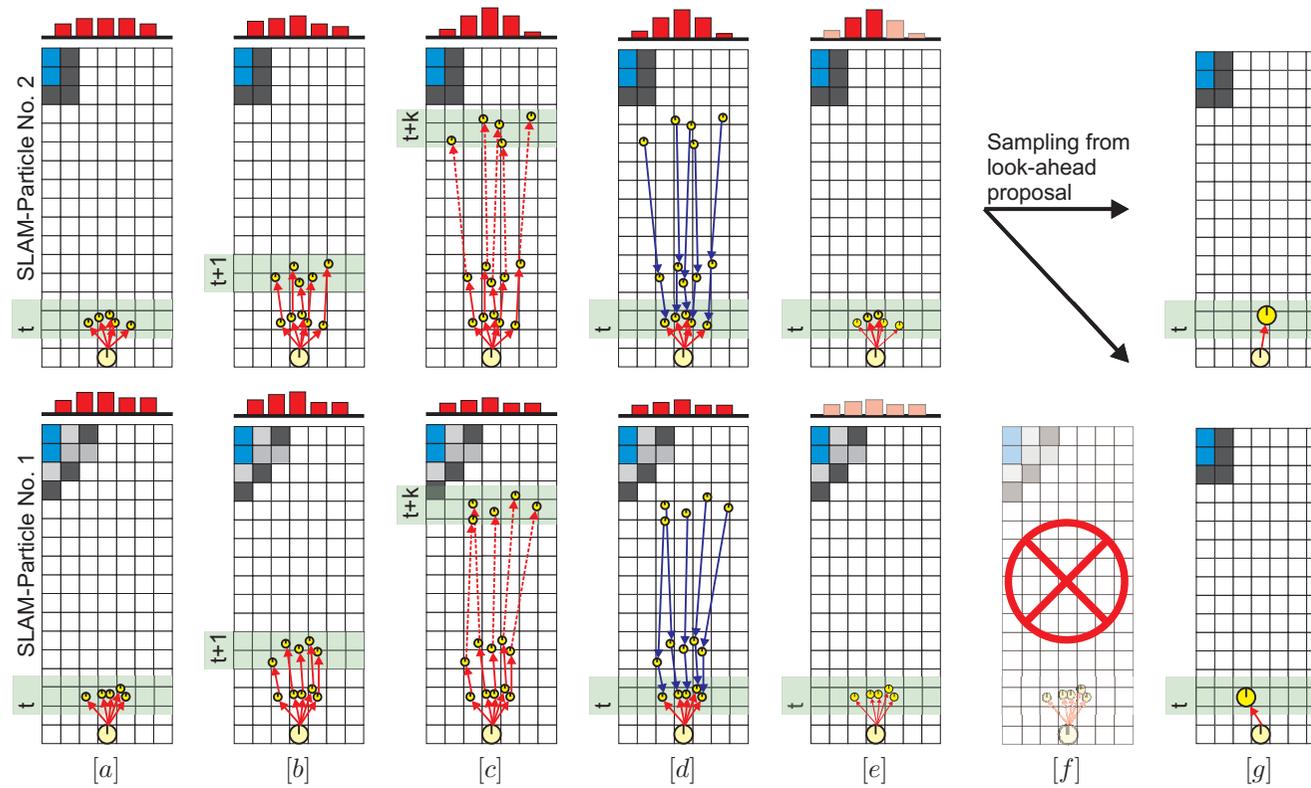


Figure 2: The algorithm for look-ahead pose sampling. For each mapping particle at time $t - 1$ do the following steps: [a] draw M localizing particles. [b] and [c]: move them according to u_{t+i} and weight them according to z_{t+i} until $t + k$ [d]: propagate back the weights of the particles to the initial situation at time t . [e], [f], and [g]: draw the successor states according to this distribution (right diagram). The histograms above the diagrams visualize the current weights of the individual localization particles at the highlighted time index. In column [e], the weights of the particles drawn as successor states are highlighted in the histogram. Note that both new SLAM particles at time t now originate from SLAM particle no. 2.

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

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

Here, $p(\mathbf{x}_{1:t} | \mathbf{z}_{1:t}, \mathbf{u}_{1:t})$ is represented by a set of sampled trajectories. In the standard approach, these samples 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 $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{z}_{t:t+k}, \mathbf{u}_{t+1:t+k}, \mathbf{m}_{t-1})$.

All in all we can compute a k -step look-ahead proposal by performing a k -step localization using M particles for each of the original N particles. For better readability, we denote these two types of samples as *localization particles* and *SLAM particles* respectively in the remainder of this paper. The nested 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 by omitting \mathbf{m}_{t-1} as:

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

Given the poses $\mathbf{x}_t \dots \mathbf{x}_{t+k}$ and the map \mathbf{m}_{t-1} , the observations $\mathbf{z}_t \dots \mathbf{z}_{t+k}$ are independent. This allows us to rewrite the term inside the integral as

$$\begin{aligned} p(\mathbf{x}_{t:t+k} | \mathbf{x}_{t-1}, \mathbf{u}_{t:t+k}, \mathbf{z}_{t:t+k}) &= \eta \cdot p(\mathbf{z}_{t+k} | \mathbf{x}_{t-1:t+k}, \mathbf{u}_{t:t+k}, \mathbf{z}_{t:t+k-1}) p(\mathbf{x}_{t-1:t+k}, \mathbf{u}_{t:t+k}, \mathbf{z}_{t:t+k-1}) \\ &= \eta \cdot p(\mathbf{z}_{t+k} | \mathbf{x}_{t+k}) p(\mathbf{x}_{t-1:t+k}, \mathbf{u}_{t:t+k}, \mathbf{z}_{t:t+k-1}), \end{aligned} \quad (5)$$

where $\eta = p(\mathbf{x}_{t-1}, \mathbf{u}_{t:t+k}, \mathbf{z}_{t:t+k})^{-1}$. Iterating the last step over $\mathbf{z}_{t+k-1}, \dots, \mathbf{z}_t$ and $\mathbf{x}_{t+k}, \dots, \mathbf{x}_t$ leads to

$$p(\mathbf{x}_{t:t+k} | \mathbf{x}_{t-1}, \mathbf{u}_{t:t+k}, \mathbf{z}_{t:t+k}) = \eta' \prod_{\tau=t}^{t+k} p(\mathbf{z}_\tau | \mathbf{x}_\tau) \cdot \prod_{\tau=t}^{t+k} p(\mathbf{x}_\tau | \mathbf{x}_{\tau-1}, \mathbf{u}_\tau). \quad (6)$$

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 6 (the localization particles) can be obtained by sampling a sequence of poses according to the sequence of motion commands

$$\mathbf{x}_{t:t+k}^{[i]} \sim \prod_{\tau=t}^{t+k} p(\mathbf{x}_\tau^{[i]} | \mathbf{x}_{\tau-1}^{[i]}, \mathbf{u}_\tau), \quad v_{t:t+k}^{[i]} \propto \prod_{\tau=t}^{t+k} p(\mathbf{z}_\tau | \mathbf{x}_\tau^{[i]}). \quad (7)$$

A sampled approximation of the integral in Eq. 4

$$p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_{t:t+k}, \mathbf{z}_{t:t+k}) \sim \left\langle \mathbf{x}_t^{[i]}, \hat{v}_t^{[i]} \right\rangle \quad (8)$$

is recovered from the samples in Eq. 7. Here $\mathbf{x}_t^{[i]}$ and $\hat{v}_t^{[i]}$ denotes one localization particle and its weight. These quantities are computed by truncating each trajectory $\mathbf{x}_{t:t+k}^{[i]}$ at time t and by propagating the likelihoods of the observations along the trajectory backwards in time (see the forth column of Figure 2). Due to resampling operations, the evolution of these trajectories can be described as an ancestor tree [Eliazar and Parr, 2003]. In our algorithm, the ancestor tree has M leafs, and a depth of k . Each level of the tree corresponds to a step of the localization algorithm. The parent of a particle in the tree at level i is the particle of level $i - 1$ from which the particle at level i has been re-sampled.

To compute the weight of the sample $\mathbf{x}_t^{[i]}$ one has to traverse the tree from the leafs at time $t + k$ up to the root at time t by summing up the weights of the leafs. Thus, the sample $\mathbf{x}_t^{[i]}$ receives a weight $\hat{v}_t^{[i]}$ which is the sum of the weights of its successors at time $t + k$

$$\hat{v}_t^{[i]} = \sum_{h=1}^M \delta_{ih} \cdot v_{t:t+k}^{[h]}, \quad (9)$$

$$\delta_{ih} = \begin{cases} 1 & \text{if particle } i \text{ at time } t \text{ is parent of particle } h \text{ at time } t+k \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

The set $\mathcal{S} = \left\{ \left\langle \mathbf{x}_t^{[i]}, \hat{v}_t^{[i]} \right\rangle \right\}$ is a sampled representation of our proposal distribution. We can draw from this set N new SLAM particles $\{\mathbf{x}_t^{[j]}\}$ for time t according to the importance weights $\hat{v}_t^{[i]}$. Note that \mathcal{S} contains the $N \cdot M$ localization particles of *all* SLAM particles.

According to the importance sampling principle (see Eq. (2)), an approximation of the true posterior $p(\mathbf{x}_t^{[j]} | \mathbf{x}_{t-1}^{[j]}, \mathbf{z}_t, \mathbf{u}_t)$ is recovered from this set by assigning to each newly drawn sample $\mathbf{x}_t^{[j]}$ a weight $w_t^{[j]}$ that corrects the bias introduced by the proposal distribution:

$$w_t^{[j]} \propto w_{t-1}^{parent(j)} \frac{p(\mathbf{x}_t^{[j]} | \mathbf{x}_{t-1}^{[j]}, \mathbf{z}_t, \mathbf{u}_t)}{p(\mathbf{x}_t^{[j]} | \mathbf{x}_{t-1}^{[j]}, \mathbf{u}_{t:t+k}, \mathbf{z}_{t:t+k})} \quad (11)$$

$$\propto w_{t-1}^{parent(j)} \frac{p(\mathbf{z}_t | \mathbf{x}_t^{[j]}) p(\mathbf{x}_t^{[j]} | \mathbf{x}_{t-1}^{[j]}, \mathbf{u}_t)}{p(\mathbf{x}_t^{[j]} | \mathbf{x}_{t-1}^{[j]}, \mathbf{u}_{t:t+k}, \mathbf{z}_{t:t+k})} \quad (12)$$

$$\propto w_{t-1}^{parent(j)} \frac{p(\mathbf{z}_t | \mathbf{x}_t^{[j]})}{\hat{v}_t^{[j]}}. \quad (13)$$

The last step follows directly from Eq. (8) and the fact that the trajectories $\mathbf{x}_{t:t+k}^{[i]}$ have been drawn by 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 \hat{v}_t at time $t + k$. Both values are readily available from the localization run.

Note that our algorithm updates the filter in a sliding window-like manner. Despite of this, we do not integrate information multiple times, since the “discounting” of particle weights by $\hat{v}_t^{[j]}$ as outlined above effectively removes the measurement information between $t + 1$ to $t + k$ from the distribution. Since the weight correction is performed *after* the new set of SLAM particles has been drawn, the presented approach does not change the estimated posterior distribution, but only the way it is represented by the limited number of particles in 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. We compare our approach using k -step look-ahead proposals to a state-of-the-art technique [Grisetti *et al.*, 2007], which uses a dynamically adapted proposal based on scan matching and to a standard RBPF mapper using odometry-based proposals. The results show that our approach is more robust in poorly structured environments, while also providing high quality maps in typical, highly-structured environments. For better readability, we use the following abbreviations to denote the alternative mapping approaches:

OP: Standard RBPF mapper with *odometry-based* proposal

SMP: RBPF mapper with one-step look-ahead using laser *scan-matching* as proposed in [Grisetti *et al.*, 2007]

LP: Our RBPF mapper using k -step *look-ahead* proposals

For *LP*, we give results for different look-ahead lengths k . It should be noted that for the case $k = 1$, our algorithm uses the same information in each iteration as *SMP*, i.e., the most recent laser scan \mathbf{z}_t . The difference lies in the way, in which this observation is used to construct the proposal distribution. By relying on local laser scan-matching, *SMP* is biased toward the most-likely alignment of the current scan to the existing map. The approach thus shows excellent performance in highly structured environments but degrades when only little structure is available. By performing sampling-based localization instead, our approach is not biased in this way, but rather depends on having a sufficiently high number of localization particles available.

Measuring Map Quality Assessing the quality of occupancy grid maps is a non-trivial problem. Straight-forward measures, like the absolute mean squared error of the corrected trajectory relative to an assumed true one, are overly sensitive to distortions on a global level. Moderate distortions, however, do typically not lead to practical problems. The same argument holds for correlation-based measures applied directly to the constructed grid-maps. Many works have thus applied measures of *relative* deviation from a ground-truth trajectory. While

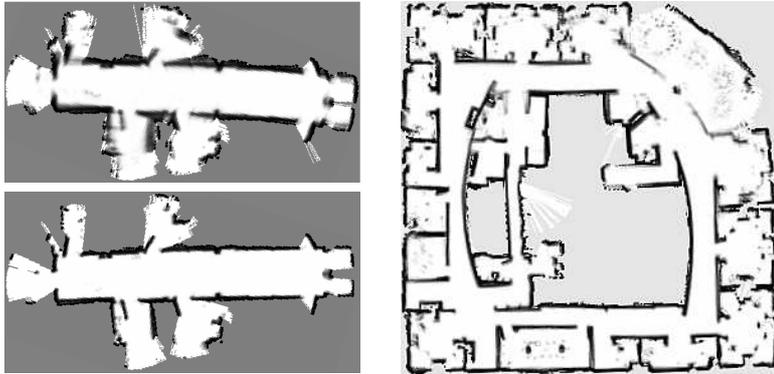


Figure 3: Mapping an office environment using *OP* (upper left) and our *LP* approach (lower left). Map of the Intel Building using our approach (right) with 50 mapping particles, 50 localization particles per map, and a look-ahead of 6. This scenario requires closing one large loop.

this is a more suitable measure in realistic scenarios, it still requires the availability of ground-truth trajectories, and it sometimes fails to capture consistency problems in the estimated maps.

We thus define a measure denoted as the *revisiting error* (*RE*), 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 and does not require ground truth information. For calculating the *revisiting error* in an experiment, we add color markers to the ground at places that are to be traversed several times during the experiment. We then record the timestamps at which the robot passes over these checkpoints. Let p be a checkpoint visited at times t_1 and t_2 . The *RE* for this location is defined as

$$\epsilon_{t_1, t_2}^p = \sum_j w^{[j]} \sqrt{(1 - \lambda)(\Delta x_j^2 + \Delta y_j^2) + \lambda \Delta \theta_j^2},$$

with $\Delta x_j := x_{t_1}^{[j]} - x_{t_2}^{[j]}$, and $\Delta y_j, \Delta \theta_j$ defined analogously. Here, x, y , and θ are the components of the pose vector, $\lambda \in [0, 1]$ is a weighing factor for the rotational component, and j denotes the index of the SLAM particles (map hypotheses) that constitute the posterior distribution over maps. Intuitively, the revisiting error for a given checkpoint is the (weighted) distance between the estimated positions within every map, weighted by the corresponding map-probability.

Mapping Highly-structured Office Environments We tested our approach in the office environment depicted in the left-hand side diagrams of Figure 3. Given a log-file recorded using a ActivMedia Pioneer II robot in this environment, we compared our approach *LP* to *SMP* and *OP* in terms of the

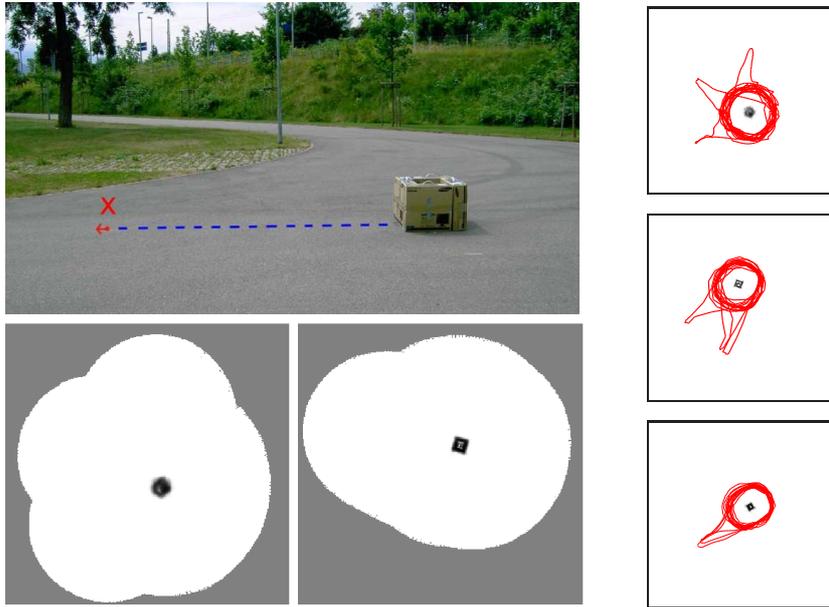


Figure 4: Mapping a low-structured environment (top left). The robot starts at location X , revolves around the box several times and, during this, returns to location X three times. Maps obtained by OP (bottom left) and our approach LP (bottom center). Corrected trajectories of the robot using OP (top right), SMP (middle right) and our LP approach (bottom right). All approaches used 50 SLAM particles. Our approach used 100 localization particles per map and a look-ahead of $k = 5$.

revisiting accuracy measure defined above. The two diagrams depict the maps obtained using OP (upper image) versus LP (lower image) with 20 mapping particles. In our approach, we used 50 localization particles per SLAM particle and a look-ahead of $k = 3$. The map generated using the SMP approach is nearly indistinguishable to ours. The revisiting accuracy is between 5 cm and 10 cm for OP and less than 5 cm for both, LP and SMP .

We also tested our algorithm using the freely available *Intel* data set to demonstrate that our approach is able to close loops in larger environments. The resulting map for 50 SLAM particles, 50 localization particles per map, a look-ahead of $k = 6$, and a maximum laser range of 9 meters is depicted in the right diagram of Figure 3. As can be seen from the diagram, we obtain a consistent map (i.e. no double walls) although only 50 SLAM particles are available to represent the uncertainty about the map. Note that approaches based on scan-matching typically lead to more accurate maps in such an environment, since enough structural information is available to precisely align the scans.

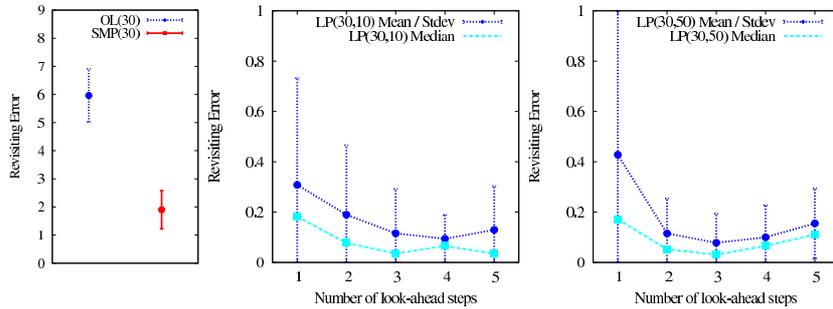


Figure 5: Revisiting error of *OP* and *SMP* in the low-structured environment with 30 SLAM particles (left), our *LP* approach with 30 SLAM particles, 10 localization particles per map (middle), and *LP* with 30 SLAM particles, 50 localization particles per map (right). The error is given in terms of mean, standard deviation, and median over a sample set of 25 runs.

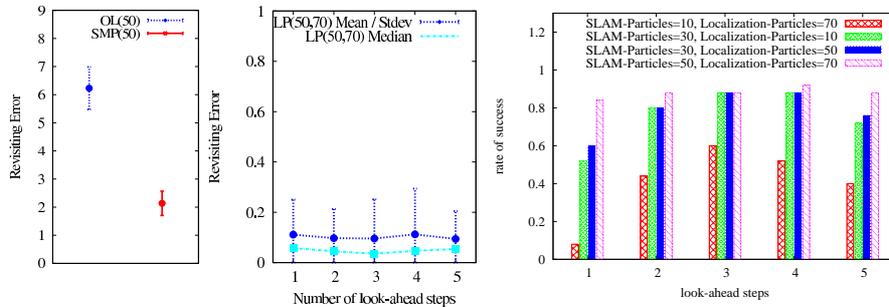


Figure 6: Revisiting error of *OP* and *SMP* in the low-structured environment with 50 SLAM particles (left), our *LP* approach with 50 SLAM particles, 70 localization particles per map (middle). Success-rates of *LP* for different parameter settings (success = revisiting error less than 0.2, right diagram).

Mapping Low-Structured Outdoor Environments We evaluated the three alternative mapping approaches in the outdoor environment depicted in the upper left picture of Figure 4. An ActivMedia Pioneer II robot started at the checkpoint X and moved around the cardboard box 15 times. During this, it returned to the checkpoint every 5th run. The maximum range of its on-board Sick Laser Range finder was limited to 2m, such that the cardboard box was the only visible map element. The distance between the checkpoint X and the box is approximately 4m.

The maps and robot trajectories in Figure 4 depict typical results taken from 25 runs of the *OP*, *SMP*, and *LP* approaches. As can be seen from the diagrams, *OP* fails to map this environment and yields a highly inconsistent trajectory (top right). Although, when using *SMP*, the map quality is higher and the box has retained its squared shape (similar to the *LP* map depicted in middle diagram

on bottom of the figure), the resulting trajectory is still inconsistent (middle right). Our approach, in contrast, is able to deal with this hard situation and produces a precise map as well as a consistently corrected trajectory (bottom right).

To quantitatively evaluate the performance of the approaches for different parameter settings, we analyzed the revisiting error (RE, see above) for different numbers of particles and a varying amount of look-ahead steps. Each approach was executed 25 times for each parameter setting. We measured ϵ_{t_1, t_2}^X , with t_1 and t_2 being the timestamps, when the robot passed the checkpoint X for the first and the last time respectively. Figure 5 and Figure 6 give the mean revisiting errors, standard deviations, and the median errors for the 25 runs and the different parameter settings using our three algorithms.

The right diagram of Figure 6 gives the success rates of *LP*, i.e., the ratio of successful runs w.r.t. the total number of runs. We count a run as successful, if its revisiting error ϵ is lower than 0.2. Visually speaking, we expect the end pose of the robot to be no more than 20cm off its starting location in the final posterior of the filter. Given a grid map resolution of 5cm, this corresponds to a deviation of four cells. Note that due to the limited range of 2m, neither *OP* nor *SMP* were able to generate consistent maps.

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 mapping systems [Grisetti *et al.*, 2007, Hähnel *et al.*, 2003]. Although the *SMP* mapper outperforms the standard RBPF in less structured environments, it achieves significantly less accurate maps than the technique proposed in this paper. By sampling k steps ahead in time, our approach is basically able to “look through gaps” of low structure until enough structure is provided for proper localization. In this way, it can map sparse environments even with relatively small sets of particles.

As can be seen from Figure 5 and 6, the revisiting error reaches an optimum for a certain number k of look-ahead steps and does not decrease monotonically for growing k . This is due to the fact that long look-ahead sequences lead the localization particles into unknown map areas. In mostly unknown areas of the map, in turn, it is hard to correctly characterize the distribution of possible observations. Thus, no state-of-the-art observation model used in grid-based SLAM can avoid introducing a bias to the distribution of particle weights in unknown areas, which leads to a degrading filter performance at some point. In the literature, this is discussed as one of the open problems in mapping, localization, and exploration. For a given practical application, the optimal number of look-ahead steps can be found in the same manner as other system parameters, e.g., the number of particles.

Due to the k -step look-ahead procedure, the filter is effectively delayed by k time steps. For real-time control (e.g., in exploration tasks), one would thus chain the odometry measurements $u_{t+1:t+k}$ to the position currently estimated by the SLAM algorithm to estimate the current location. Alternatively, control could be based on the current posterior of the localization particles. Or, if at any point in time, high precision and reliability are required, the k -step look-

ahead “buffer” can be flushed in a straight-forward manner to integrate all available information. In our experiments, even the more difficult situations did not require more than 10 look-ahead steps, which corresponds to a delay in the order of 1-2 seconds.

The computational complexity of the look-ahead step is proportional to the total number of localization particles $M \cdot N$, the number of look-ahead steps k and to the computational cost of evaluating the likelihood of a particle $p(\mathbf{z} | \mathbf{x})$. The expensive part of the algorithm is to compute the likelihood of the localization particles, as it is the case for most particle filter-based algorithms. This step can be made efficient, e.g., by using the likelihood fields sensor model for range sensors [Thrun *et al.*, 2005]. In this way, computing the likelihood of one particle requires h accesses to a lookup table, where h is the number of laser beams used. Our implementation computes the likelihoods of 2500 particles using 90 laser beams in around 10 *ms* on a 2 GHz Dual-Core laptop. Additionally, when using look-ahead proposals, one typically requires less SLAM particles for mapping. This reduces the memory requirements of the filter.

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 k -step look-ahead proposal distribution to yield a more informed pose estimate. 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 data sets. We furthermore compared our technique to two state-of-the-art mapping techniques. Experimental results suggest that our approach yields highly accurate maps and outperforms alternative approaches in environments with sparse features.

Acknowledgments

This work has partly been supported by the EC under contract number FP6-IST-034120 and under contract number FP6-004250-CoSy, by the German Science Foundation (DFG) under contract number SFB/TR-8 (A3) and the Federal Ministry of Education and Research (BMBF) under contract number 01IMEO1F.

References

- [Beevers and Huang, 2007] K. Beevers and W. Huang. Fixed-lag sampling strategies for particle filter slam. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2007.

- [Doucet *et al.*, 2000] A. Doucet, J.F.G. de Freitas, K. Murphy, and S. Russell. Rao-Blackwellized particle filtering for dynamic bayesian networks. In *Proc. of the Conf. on Uncertainty in Artificial Intelligence (UAI)*, pages 176–183, Stanford, CA, USA, 2000.
- [Doucet *et al.*, 2001] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte-Carlo Methods in Practice*. Springer Verlag, 2001.
- [Eliazar and Parr, 2003] A. Eliazar and R. Parr. DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks. In *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, pages 1135–1142, Acapulco, Mexico, 2003.
- [Fox *et al.*, 1999] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research (JAIR)*, 11:391–427, 1999.
- [Grisetti *et al.*, 2007] 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.
- [Hähnel *et al.*, 2003] 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)*, Las Vegas, NV, USA, 2003.
- [Howard, 2005] Andrew Howard. Multi-robot simultaneous localization and mapping using particle filters. In *Robotics: Science and Systems*, Cambridge, MA, USA, 2005.
- [Montemerlo *et al.*, 2002] 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.
- [Montemerlo *et al.*, 2003] 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.
- [Murphy, 1999] 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.
- [Pitt and Shephard, 1997] M.K. Pitt and N. Shephard. Filtering via simulation: auxiliary particle filters. Technical report, Department of Mathematics, Imperial College, London, 1997.

- [Stachniss *et al.*, 2007] C. Stachniss, G. Grisetti, W. Burgard, and N. Roy. Analyzing gaussian proposal distributions for mapping with rao-blackwellized particle filters. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2007.
- [Thrun *et al.*, 2005] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.
- [Tipaldi *et al.*, 2007] G.D. Tipaldi, A. Farinelli, L. Iocchi, and D. Nardi. Heterogeneous feature state estimation with rao-blackwellized particle filters. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 3850–3855, 2007.