Fast and Accurate SLAM with Rao-Blackwellized Particle Filters

Giorgio Grisetti [a,b] Gian Diego Tipaldi [b] Cyrill Stachniss [c,a]
Wolfram Burgard [a] Daniele Nardi [b]

[a] *University of Freiburg, Department of Computer Science, D-79110 Freiburg, Germany*

[b] *Dipartimento Informatica e Sistemistica, Universitá "La Sapienza", I-00198 Rome, Italy*

[c] *Swiss Federal Institute of Technology (ETH) Zurich, CH-8092 Zurich, Switzerland*

Abstract

Rao-Blackwellized particle filters have become a popular tool to solve the simultaneous localization and mapping problem. This technique applies a particle filter in which each particle carries an individual map of the environment. Accordingly, a key issue is to reduce the number of particles and/or to make use of compact map representations. This paper presents an approximative but highly efficient approach to mapping with Rao-Blackwellized particle filters. Moreover, it provides a compact map model. A key advantage is that the individual particles can share large parts of the model of the environment. Furthermore, they are able to reuse an already computed proposal distribution. Both techniques substantially speed-up the overall filtering process and reduce the memory requirements. Experimental results obtained with mobile robots in large-scale indoor environments and based on published standard datasets illustrate the advantages of our methods over previous mapping approaches using Rao-Blackwellized particle filters.

Key words: SLAM, Rao-Blackwellized particle filter, grid map, informed proposal
PACS:

1 Introduction

Learning maps is a fundamental task of mobile robots and a lot of researchers focused on this problem. In the literature, the mobile robot mapping problem is often referred to as the *simultaneous localization and mapping (SLAM)* problem [1, 2, 3, 4, 5, 6, 7, 8]. In general, SLAM is a complex problem because for learning a map the robot requires a good pose estimate while at the same time a consistent map is needed to localize the robot. This dependency between the pose and the map estimate makes the SLAM problem hard and requires to search for a solution in a high-dimensional space.

Murphy, Doucet, and colleagues [7, 9] introduced Rao-Blackwellized particle filters (RBPFs) as an effective means to solve the SLAM problem. The main problem of Rao-Blackwellized particle filters lies in their complexity, measured in terms of the number of particles required to learn an accurate map. Reducing this quantity is one of the major challenges for this family of algorithms.

The contribution of this paper is a technique that reduces the computational and the memory requirements in the context of mapping with Rao-Blackwellized particle filters. In this way, it becomes feasible to maintain a comparably large set of particles online. This is achieved by enabling a subset of samples to share large parts of the map and to use the same proposal distribution. Our system allows a standard laptop computer to perform all computations necessary to learn accurate maps with more than one thousand samples online.

This paper is organized as follows. After the discussion of related work, we briefly introduce mapping with RBPFs. We then describe our technique for efficiently drawing particles from a proposal distribution. After this, we present our map representation and the concept of particle clusters. Finally, we show experiments illustrating the improvements of our approach to map learning with RBPFs.

2 Related Work

The estimation techniques for the SLAM problem can be classified according to their underlying basic principle. The most popular approaches are extended Kalman filters (EKFs), maximum likelihood techniques, sparse extended information filters (SEIFs), and Rao Blackwellized particle filters (RBPFs). The effectiveness of the EKF approaches comes from the fact that they estimate a fully correlated posterior over landmark maps and robot poses [10, 11]. Their weakness lies in the strong assumptions that have to be made on the robot motion model and the sensor noise. Moreover, in the basic framework the landmarks are assumed to be uniquely identifiable. There exist techniques [12] to deal with unknown data association in the SLAM context, however, if certain assumptions are violated, the filter is likely to diverge [13].

Thrun *et al.* [8] proposed a SEIF method which is based on the inverse of the covariance matrix. In this way, measurements can be integrated efficiently. Eustice *et al.* [14] presented an improved technique to accurately compute the error-bounds within the SEIF framework and thus reduces the risk of becoming overly confident. Paskin [15] presented a solution to the SLAM problem using thin junction trees. This reduces the complexity compared to EKF-based approaches since thinned junction trees provide a linear-time filtering operation.

An alternative approach is to use a maximum likelihood algorithm that computes a map by constructing a network of relations. The relations represent the spatial

constraints between the poses of the robot [3, 16]. The main difference to RBPFs is that the maximum likelihood approach can only track a single mode of the distribution about the trajectory of the robot. It computes the solution by minimizing the least square error introduced by the constraints.

Lisien *et al.* [17] realized an hierarchical map model in the context of SLAM and reported that this improves loop-closing. Bosse *et al.* [18] describe a generic framework for SLAM in large-scale environments. They use a graph structure of local maps with relative coordinate frames similar to the work described in [19]. This approach is able to reduce the complexity of the overall problem and it better deals with the linearizations in the context of EKF-SLAM. Our approach is related to this framework since we also use local maps attached to a graph structure to model the environment. However, our motivation to use such a map representation is to allow multiple particles to share local maps and to compute the proposal distributions in an efficient way.

Murphy [7] introduced Rao-Blackwellized particle filters as an effective means to solve the SLAM problem. Each particle in a RBPF represents a potential trajectory of the robot and a map of the environment. The framework has been subsequently extended by Montemerlo *et al.* [5, 6] for approaching the SLAM problem with landmarks. To learn accurate grid maps, Hähnel *et al.* [4] presented an improved motion model that reduces the number of required particles. A combination of the approach of Hähnel *et al.* and Montemerlo *et al.* as been presented by Grisetti *et al.* [2], which extends the ideas of FastSLAM-2 [5] to the grid map case. We present in this paper an approximative solution to RBPF-based mapping which describes how to draw particles and how to represent the maps of the particles so that the system can be executed significantly faster and needs less memory resources.

There exist other approaches to mapping with RBPFs like DP-SLAM [1] that provide a compact map representation. This approach stores an ancestry tree of particles. Furthermore, each cell of their grid map maintains a tree of poses from which that cell has been observed. This allows the system to store the map hypotheses in an compact manner. Additionally, the resampling can be carried out more efficiently. In contrast to that, our map representation enables us to reuse already computed proposal distributions for multiple samples. This is done by carrying out a coordinate transformation between the reference frames stored in our graph structure.

The contribution of this paper is a computational and memory efficient Rao-Blackwellized particle filter for SLAM. Our approach allows the robot to efficiently determine the proposal distributions to sample the next generation of particles in an approximative manner. Additionally, we present a compact map model in which multiple particles share local maps. This enables us to maintain substantially more samples with less memory and computational requirements compared to state-of-the-art mapping approach using Rao-Blackwellized particle filters.

3 Learning Maps with Rao-Blackwellized Particle Filters

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 trajectory $x_{1:t} = x_1, \ldots, x_t$ of the robot and the map m of the environment given the observations $z_{1:t} = z_1, \ldots, z_t$ and odometry measurements $u_{1:t-1} = u_1, \ldots, u_{t-1}$. It does so by using the following factorization:

$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t-1}) = p(x_{1:t} \mid z_{1:t}, u_{1:t-1}) p(m \mid x_{1:t}, z_{1:t}) \tag{1}$$

In this equation, the posterior $p(x_{1:t} \mid z_{1:t}, u_{1:t-1})$ is similar to the localization problem, since only the trajectory of the vehicle needs to be estimated. This estimation is performed using a particle filter which incrementally processes the observations and the odometry readings as they are available. The second term $p(m \mid x_{1:t}, z_{1:t})$ can be computed efficiently since the poses $x_{1:t}$ of the robot are known when estimating the map m. Therefore, a Rao-Blackwellized particle filter for SLAM maintains an individual map for each sample and updates this map based on the trajectory estimate of the sample upon "mapping with known poses".

A mapping system that applies a RBPF requires a proposal distribution in order to draw the next generation of samples. The general framework leaves open which proposal should be used and how it should be computed. A proposal distribution typically used in the context of Monte-Carlo localization is the motion model $p(x_t \mid x_{t-1}, u_{t-1})$. This proposal, however, is sub-optimal since it does not consider the observations of the robot to predict its motion. As pointed out by several authors [20, 5], problem-specific proposal distributions are needed in order to build an efficient mapping system. The approach presented in this paper, makes use of our previously defined [2] proposal distribution. It transfers the ideas of FastSLAM-2 [5] to the grid map case. Under the Markov assumption, the optimal proposal distribution [20] is

$$p(x_t \mid m_{t-1}^{(i)}, x_{t-1}^{(i)}, z_t, u_{t-1}) = \frac{p(z_t \mid m_{t-1}^{(i)}, x_t) p(x_t \mid x_{t-1}^{(i)}, u_{t-1})}{\int p(z_t \mid m_{t-1}^{(i)}, x') p(x' \mid x_{t-1}^{(i)}, u_{t-1}) \, dx'}. \tag{2}$$

Whenever a laser range finder is used, one can observe that the observation likelihood $p(z_t \mid m_{t-1}, x_t)$ is much more peaked than the motion model $p(x_t \mid x_{t-1}, u_{t-1})$. The observation likelihood dominates the product in Eq. (2) in the meaningful area of the distribution. Therefore, we approximate $p(x_t \mid x_{t-1}, u_{t-1})$ by a constant k within this meaningful area $L^{(i)}$. Under this approximation, the proposal turns into

$$p(x_t \mid m_{t-1}^{(i)}, x_{t-1}^{(i)}, z_t, u_{t-1}) \simeq p(z_t \mid m_{t-1}^{(i)}, x_t) \cdot \left[\int_{x' \in L^{(i)}} p(z_t \mid m_{t-1}^{(i)}, x') \, dx' \right]^{-1}. \tag{3}$$

Eq. (3) can be computed by evaluating $p(z_t \mid m_{t-1}^{(i)}, x_t)$ on a grid which is bounded by the maximum odometry error. Alternatively, one can use a set of sampled points

$\{x_j\}$ and then evaluate point-wise the observation likelihood. In order to efficiently sample the next generation of particles, one can approximate this distribution by a Gaussian. For each particle i, the parameters $\mu_t^{(i)}$ and $\Sigma_t^{(i)}$ of the Gaussian are computed as

$$\mu_t^{(i)} = \frac{1}{\eta^{(i)}} \sum_{j=1}^{K} x_j \cdot p(z_t \mid m_{t-1}^{(i)}, x_j) \tag{4}$$

$$\Sigma_t^{(i)} = \frac{1}{\eta^{(i)}} \sum_{j=1}^{K} p(z_t \mid m_{t-1}^{(i)}, x_j) \cdot (x_j - \mu_t^{(i)})(x_j - \mu_t^{(i)})^T. \tag{5}$$

Here $\eta = \sum_{j=1}^{K} p(z_t \mid m_{t-1}^{(i)}, x_j)$ is a normalizer. Note that $\mu_t^{(i)}$ and $\Sigma_t^{(i)}$ are calculated for each particle individually which is computationally expensive but leads to an informed proposal distribution. This allows us to draw particles in an more accurate manner which seriously reduces the number of required samples.

4 Speeding Up the Computation of the Proposal

The problem of the method presented above is the computational complexity of the informed proposal distribution since it has to be done for each sample individually. As a result, such a mapping system runs online only for small particle sets. Furthermore, each particle maintains a full grid map which requires to store large grid structures in the memory. To overcome this limitation, we present a way to utilize intermediate results in order to efficiently determine the proposal for the individual samples. Our implementation extends the open-source implementation [21] of the mapping system of Grisetti *et al.* [2] which originally makes use of the proposal distribution presented in the previous section.

The proposal distribution is needed to model the relative movement of the vehicle under uncertainty. In most situations, this uncertainty is similar for all samples within one movement. It therefore makes sense to use the same uncertainty to propagate the particles. We derive a way to sample multiple particles from the same proposal. As a result, the time consuming computation of the proposal distribution can be carried out for a few particles that are representatives for groups of similar samples.

Furthermore, we observed that local maps which are represented in a particle-centered coordinate frame look similar for many samples. We therefore present a compact map model in which multiple particles can share their local maps. Instead of storing a full grid map, each sample maintains only a set of reference frames for the different local maps. This substantially reduces the memory requirements of the mapping algorithm.

4.1 Different Situations During Mapping

Before we derive our new proposal distributions, we start with a brief analysis of the behavior of a RBPF. One can distinguish three different types of situations during mapping:

- The robot is moving through *unknown* areas,
- is moving through *known* areas, or
- is *closing a loop*. Here, closing a loop means that the robot first moves through unknown areas and then reenters known terrain. It can be seen as moving along a so far non traversed shortcut from current pose of the robot to an already known area (see also [22]).

In each of those situations, the filter behaves differently. Whenever the robot is moving through unknown terrain, the uncertainty about the pose of the robot grows. This is due to the fact that the errors are accumulated along the trajectory. The resulting uncertainty can only be bounded by observations which cover a (partially) known region.

In the second case, a map of the surroundings of the robot is known and in this way the SLAM problem turns into a localization problem which is typically easier to handle. Whenever the robot is closing a loop, the particle cloud is often widely spread. By reentering known areas, the filter can typically determine which particles are consistent with their own map and which are not. As a result, such a situation leads to an unbalanced distribution of particle weights. The next resampling action then eliminates a series of unlikely hypotheses and the uncertainty decreases.

For each of these three situations, we will present a proposal distribution that needs to be computed only for a small set of representatives rather than for all particles. Throughout this paper, we make the following three assumptions.

Assumption 1 The current situation is known, which means that the robot can determine whether it is moving through unknown terrain, within a known area, or is closing a loop.

Assumption 2 The corresponding local maps of two samples are similar if considered in a particle-centered reference frame. In the following, we refer to this property as *local similarity* of the maps.

Assumption 3 An accurate algorithm for pose tracking is used and the observations are affected by a limited sensor noise.

4.2 Computing the Proposal for Unknown Terrain

For proximity sensors like laser range finders, the observations of the robot cover only a local area around the robot. As a result, we only need to consider the sur-

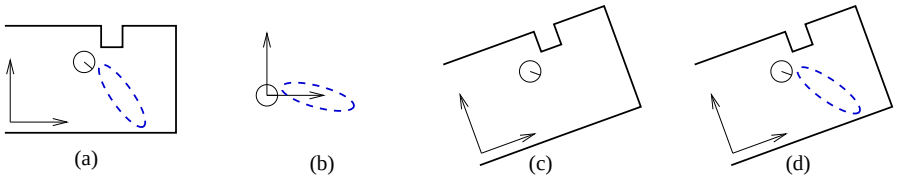

Figure 1. Image (a) depicts the pose of a particle, its local map, and the computed proposal which represented by the blue/dashed ellipse. Image (b) illustrates the proposal distribution represented in the ego-centric reference frame of that sample. Image (c) shows a second particle and its map. By carrying out a coordinate transform, the proposal of the first particle can be used by the second particle as long as their maps are (locally) similar (d).

roundings of the robot when computing the proposal distribution. Let $\tilde{m}_{t-1}^{(i)}$ refer to the local map of particle i around its previous pose $x_{t-1}^{(i)}$. In the surroundings of the robot, we can approximate

$$p(x_t \mid m_{t-1}^{(i)}, x_{t-1}^{(i)}, z_t, u_{t-1}) \;\simeq\; p(x_t \mid \tilde{m}_{t-1}^{(i)}, x_{t-1}^{(i)}, z_t, u_{t-1}). \tag{6}$$

Let $\oplus$ and $\ominus$ be the standard pose compounding operators (see [16]): $a \ominus b$ is an operator that translates all the points in the domain of the function a so that the new origin of the domain of a is b and $\oplus$ is its inverse. The local similarity between maps (Assumption 2) allows us to write $\tilde{m}_{t-1}^{(i)} \ominus x_{t-1}^{(i)} \simeq \tilde{m}_{t-1}^{(j)} \ominus x_{t-1}^{(j)}$. In this case, the proposal distribution for different particles are similar if transformed to an ego-centric reference frame

$$p(x_t \ominus x_{t-1}^{(j)} \mid \tilde{m}_{t-1}^{(j)}, x_{t-1}^{(j)}, z_t, u_{t-1}) \simeq p(x_t \ominus x_{t-1}^{(i)} \mid \tilde{m}_{t-1}^{(i)}, x_{t-1}^{(i)}, z_t, u_{t-1}). \tag{7}$$

As a result, we can determine the proposal for a particle j by computing the proposal in the reference frame of particle i and then translating it to the reference frame of particle j

$$p(x_t \mid \tilde{m}_{t-1}^{(j)}, x_{t-1}^{(j)}, z_t, u_{t-1}) \simeq p(x_{t-1}^{(j)} \oplus (x_t \ominus x_{t-1}^{(i)}) \mid \tilde{m}_{t-1}^{(i)}, x_{t-1}^{(i)}, z_t, u_{t-1}). \tag{8}$$

This computation is illustrated in Figure 1. It shows how to transform a proposal between particles. The complex proposal computation needs to be performed only once and can then be translated to the reference frame of the other particles.

4.3 Computing the Proposal for Already Visited Areas

Whenever the robot moves through known areas, each particle stays localized in its own map according to Assumption 3. To update the new pose of each particle while the robot moves, we choose the pose x_t that maximizes the likelihood of the observation around the pose predicted by odometry

$$x_t^{(i)} = \underset{x_t}{\operatorname{argmax}}\, p(x_t \mid \tilde{m}_{t-1}^{(i)}, x_{t-1}^{(i)}, z_t, u_{t-1}). \tag{9}$$

Analog to Eq. (6)-(8), we can express the proposal of particle j using the one of particle i. The only difference is that we do not apply the $\oplus$ and $\ominus$ operators based on the poses of the samples. Instead, the operators are applied based on the particle dependent reference frames $l^{(i)}$ and $l^{(j)}$ of the local maps. These reference frames were established whenever the robot visits the corresponding area for the first time. This results in

$$p(x_t \mid \tilde{m}_{t-1}^{(j)}, x_{t-1}^{(j)}, z_t, u_{t-1}) \simeq p(l^{(j)} \oplus (x_t \ominus l^{(i)}) \mid \tilde{m}_{t-1}^{(i)}, x_{t-1}^{(i)}, z_t, u_{t-1}). \quad (10)$$

Combining Eq. (9) and Eq. (10) leads to

$$x_t^{(j)} = \operatorname*{argmax}_{x_t} p(x_t \mid \tilde{m}_{t-1}^{(j)}, x_{t-1}^{(j)}, z_t, u_{t-1}) \quad (11)$$

$$\simeq \operatorname*{argmax}_{x_t} p(l^{(j)} \oplus (x_t \ominus l^{(i)}) \mid \tilde{m}_{t-1}^{(i)}, x_{t-1}^{(i)}, z_t, u_{t-1}) = l^{(j)} \oplus (x_t^{(i)} \ominus l^{(i)}). \quad (12)$$

Under the Assumptions 2 and 3, we can estimate the poses of all samples according to Eq. (12) (while moving through known areas). In this way, the complex computation of an informed proposal needs to be done only once.

4.4 Computing the Proposal When Closing a Loop

In contrast to the two situations described before, the computation of the proposal is more complex in case of a loop-closure. This is due to the fact that Assumption 2 (local similarity) is typically violated even for subsets of particles. Let us assume that the particle cloud is widely spread when the loop is closed. Typically, the individual samples reenter the previously mapped terrain at different locations. This results in different hypotheses about the topology of the environment and definitively violates Assumption 2. Dealing with such a situation, requires additional effort in the estimation process.

Whenever a particle i closes a loop, we consider that the map $\tilde{m}_{t-1}^{(i)}$ of its surroundings consists of two components. Let $m_{\text{loop}}^{(i)}$ refer to the map of the area, the robot seeks to reenter. Then, $m_{\text{local}}^{(i)}$ is the map constructed from the most recent measurements without the part of the map that overlaps with $m_{\text{loop}}^{(i)}$. Since those two maps are disjoint and under the assumption that the individual grid cells are independent, we can use a factorized form for our likelihood function

$$p(z_t \mid x_t, m_{\text{local}}^{(i)}, m_{\text{loop}}^{(i)}) \propto p(z_t \mid x_t, m_{\text{local}}^{(i)}) \cdot p(z_t \mid x_t, m_{\text{loop}}^{(i)}). \quad (13)$$

For efficiency reasons, we use only the local map $m_{\text{local}}^{(i)}$ to compute the proposal and do not consider $m_{\text{loop}}^{(i)}$. This procedure is valid but requires to adapt the weight computation. According to the importance sampling principle, this leads to

$$w_t^{(i)} = w_{t-1}^{(i)} \cdot \frac{p(x_t^{(i)} \mid z_t, x_{t-1}^{(i)}, m_{\text{local}}^{(i)}, m_{\text{loop}}^{(i)}, u_{t-1})}{p(x_t^{(i)} \mid z_t, x_{t-1}^{(i)}, m_{\text{local}}^{(i)}, u_{t-1})} \tag{14}$$

$$= w_{t-1}^{(i)} \cdot \frac{\eta_1^{(i)} p(z_t \mid x_t^{(i)}, m_{\text{local}}^{(i)}) p(z_t \mid x_t^{(i)}, m_{\text{loop}}^{(i)})}{\eta_2^{(i)} p(z_t \mid x_t^{(i)}, m_{\text{local}}^{(i)})} \tag{15}$$

$$= w_{t-1}^{(i)} \cdot p(z_t \mid x_t^{(i)}, m_{\text{loop}}^{(i)}) \frac{\eta_1^{(i)}}{\eta_2^{(i)}}, \tag{16}$$

where η_1 and η_2 are normalization factors resulting from Bayes' rule.

Note that the computation of the proposal in case of a loop-closure is more expensive than in the two other situations. Fortunately, loop-closing situations occur rarely. The robot has to travel through unknown and eventually known terrain for a comparably long period of time before a loop-closure can occur.

4.5 Approximative Importance Weight Computation

We observed in practical experiments that the normalizing factors η_1 and η_2 in Eq. (16) have only a minor influence on the overall weight. In order to speed up the computation of the importance weights, we approximate Eq. (16) by

$$w_t^{(i)} \simeq w_{t-1}^{(i)} \cdot p(z_t | x^{(i)}, m_{\text{loop}}^{(i)}) \tag{17}$$

in which the normalizing factors are neglected. This is significantly faster to compute and as we will demonstrate in the experiments leads to almost identical importance weights.

5 Achieving Situation Estimation, Local Similarity, and Pose Tracking

All of the derivations made in the previous section require that the robot knows whether it is moving through unknown terrain, through a previously mapped area, or is currently closing a loop (Assumption 1). Here, we describe how to distinguish the different cases. Detecting the first two situations can be done in a straightforward way by comparing the area covered by the current observation given the particle pose and the map constructed so far.

In general, it is more difficult to decide whether or not the robot is closing a loop. To detect loop closures, we apply the approach proposed by Stachniss *et al.* [22]. We use a dual representation consisting of a topologic map that models the trajectory of the vehicle and a grid map. By comparing both representations, one is able to accurately determine whether or not a robot is closing a loop.

Assumption 2 (local similarity) typically holds only up to the first loop closure but is then violated. By explicitly modeling the different topological hypotheses, it is

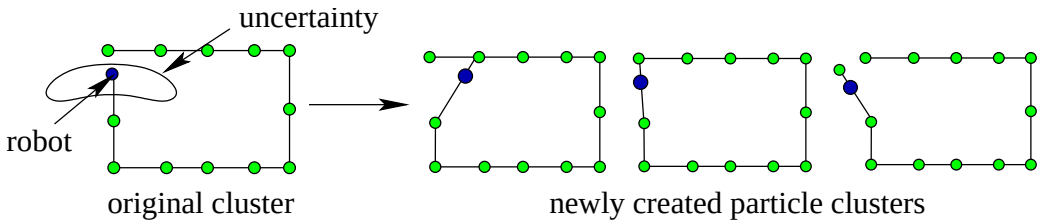

Figure 2. The left image depicts a cluster while the robot is approaching a loop-closure. The shown particle cluster splits up into three different clusters (topology hypotheses) as depicted in the right image.

still possible to represent the posterior in an appropriate way. To achieve local similarity, we introduce the notion of a *particle cluster* which describes a subset of particles for which the assumption of local similarity between maps holds. Ambiguities in the posterior can then be modeled using multiple particle clusters.

In the beginning of the mapping process, we start with a single cluster, but after closing a loop, multiple topology hypotheses typically occur. In this situation, the cluster needs to be split up. Therefore, we determine which particle belongs to which topological hypothesis in order to form new clusters. In our current implementation, we group the samples according to their Euclidian distance to the different nodes in their own graph structure of reference frames. For each particle, we first determine the list of nodes in the field of view of that sample. We order this list according to the Euclidian distance from the pose represented by the sample to the corresponding node. Then, a cluster is given by the samples which have the same list of nodes. An example which illustrates how new clusters arise in case of a loop-closure is depicted in Figure 2. Note that we currently do not merge clusters. Throughout our experiments, we observed that multiple particle clusters are created when closing a loop. The actual number ranges up to 50. However, after a few iterations only a small number of clusters (typically up to five) survive.

In our current implementation, we represent a map as a set of local maps also called patches. A global map for a given particle can be obtained by specifying the location of each patch within a global reference frame. Each sample therefore has to store only a list of reference frames $l_n^{(i)}$ for the patches. In this way, the individual patches $\mathcal{P}_1, \ldots, \mathcal{P}_N$ need to be stored only once per cluster. The map of particle i can be computed by $m^{(i)} = \bigcup_n l_n^{(i)} \oplus \mathcal{P}_n$.

Within one particle cluster, the local maps of each particle fulfills the assumption of local similarity. Therefore, they can share their patches. This results in a more compact representation compared to storing individual grid maps. In our current mapping system, we used a graph structure where each node is a reference to the corresponding patch. Furthermore, the state vector $s_t^{(i)}$ and the clusters $\mathcal{C}_k$ are rep-

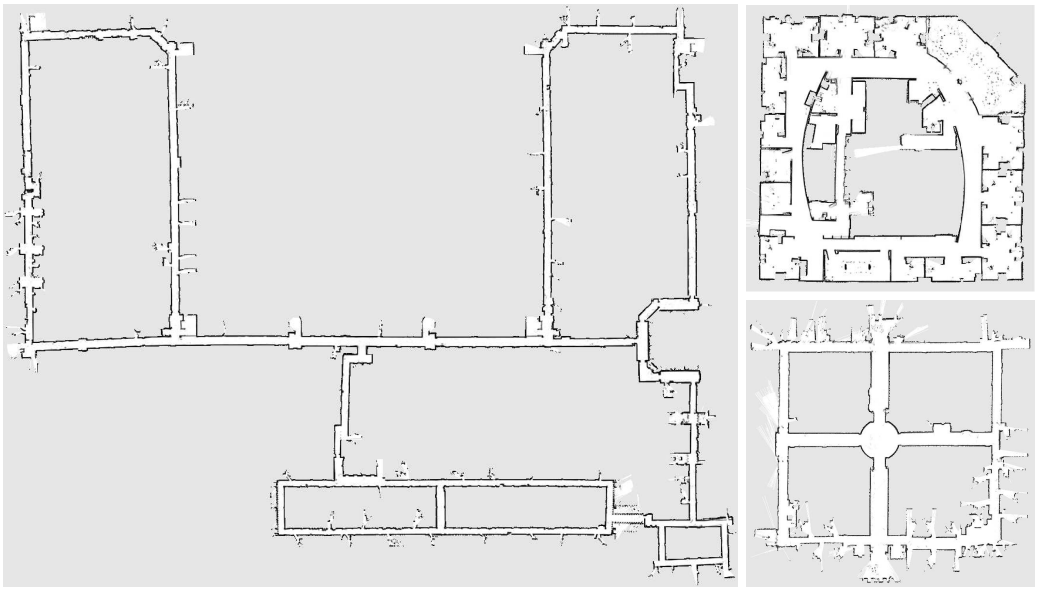

Figure 3. Learned map of the MIT Killian Court, the Intel Research lab, and the ACES dataset using our approach.

resented as

$$s_t^{(i)} = \left\langle \underbrace{x_t^{(i)}}_{\text{robot pose}} \ , \ \underbrace{k}_{\text{cluster ID}} \ , \underbrace{l_1^{(i)}, \ldots, l_{N_k}^{(i)}}_{\text{patch locations}} \right\rangle \quad \mathcal{C}_k = \left\langle \underbrace{\mathcal{P}_1, \ldots, \mathcal{P}_{N_k}}_{\text{pointers to patches}} \ , \ \underbrace{\{e_{l,m}\}}_{\text{graph edges}} \right\rangle . \tag{18}$$

Note that the number N_k of patches does not grow with the length of the trajectory traveled by the robot. It grows with the number of relevant patches which is related to the size of the environment.

To fulfill Assumption 3, we apply an incremental scan alignment technique based on laser range finder data. The experiments presented in this paper indicate that this setup/implementation is sufficient to satisfy the three assumptions. As a result, we obtain a mapping system which provides highly accurate maps in a fast and memory efficient manner.

6 Experiments

In this section, we present experiments performed on real robot datasets which are commonly used within the robotics community. In the first experiment, we corrected several log files using our approach. The left image of Figure 3 depicts the resulting map of the MIT Killian Court. This is a challenging dataset, since the environment is large (it took 2.5h to record this log file) and contains several nested loops which can rise the problem of particle depletion. As shown in the figure, the map does not contain any inconsistencies like for example double walls. Comparable results have been obtained using the Intel Research, the Austin ACES dataset, shown in the same figure.

Table 1
Comparison of memory and computational resources for the MIT dataset using a PC with a 1.3 GHz CPU.

	#particles	**execution time**	**max. memory**
our approach	2,000	51 min	210 MB
our approach	1,000	41 min	180 MB
our approach	500	30 min	165 MB
RBPF of [21]	150	(memory swapping)	2.9 GB
RBPF of [21]	80	300 min	1.5 GB
RBPF of [21]	50	190 min	1 GB

The second experiment is designed to show the advantages of our approach compared to a RBPF-based mapper without our optimizations. For this comparison, we used the open-source mapper provided in [21]. We compared the overall time, needed to correct the MIT Killian Court dataset and the memory used to store the maps. This was done using a (comparably slow) PC with a 1.3 GHz CPU and 1.5 GB RAM. The results of both mapping approaches are summarized in Table 1. In our current implementation, the filter update for *each cluster* takes in average 20 ms when moving through known terrain and 200 ms when moving through unknown terrain. When actually closing a loop, *each particle* requires approximatively 2 ms of execution time while the check for the closure takes around 0.3 ms per sample.

Since the approximated proposal is not as accurate as the original one, we need more particles to achieve the same robustness in filter convergence and quality of the resulting maps. However, we can maintain more than one order of magnitude more particles while requiring less runtime and memory. In all our experiments, this sufficiently accounted for the less accurately drawn samples.

The savings on runtime are mainly caused by transforming an already computed proposal distribution so that it can be used for several particles instead of computing it from scratch each time. The memory savings are due to the fact that all particles within a cluster can share a their local grid maps. Furthermore, the memory usage and runtime of our approach grows comparably slow when increasing the number of particles. The reason is that the complexity of our filter grows mainly with the number of topological hypotheses (particle clusters) which need to be maintained and only indirectly with the number of samples. Note that the *maximum* memory required by our approach is considerably higher than the amount of memory typically used. There exist a few peaks in the memory usage which arise from a loop closure where several clusters are temporarily created but deleted after a few steps (compare Figure 4). The typical memory usage is around 20% of the maximum usage.

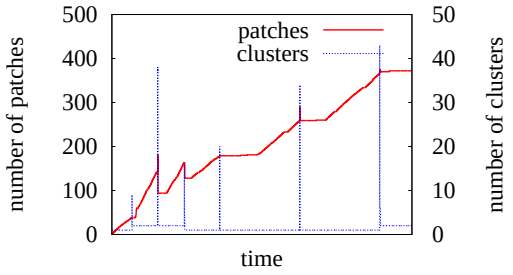

Figure 4. This plot depicts the number of patches in the memory and the number of clusters over time for the MIT dataset using 1.500 particles.

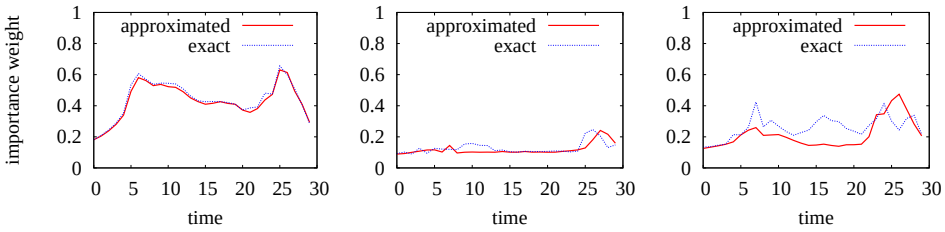

Figure 5. Difference in the particle weights caused the approximative computation for three different samples during a loop-closure. The left and middle image show typical results, the right one depicts the one of the worst results during our experiments.

Figure 4 depicts the number of patches that need to be stored and the number of clusters during the estimation process of the MIT dataset with 1,500 particles. As can be seen, the number of clusters is typically small until a loop closure occurs. At this point, the number of clusters increases. However, after a short period of time most of the clusters vanish.

The last experiment evaluates the error introduced by our approximative importance weight computation when closing a loop. As presented in Eq. (17), we ignore the normalization factors to achieve a faster estimation. We analyzed the loop-closing actions and in most situations the approximation error was small. Figure 5 depicts the differences between the sound computation and our approximation for three different particles during a loop closure. For a more quantitative evaluation between both methods, we computed the KL-divergence (KLD) between the distribution of the importance weights in both cases. It turned out, that the average KLD was only 0.02 (a KLD of 0 means that the distributions are equal and the higher the value the more different are the distributions). Substantiated by the good approximation quality, we ignore the evaluation of η_1 and η_2 when computing the particle importance weight.

7 Conclusion

In this paper, we presented efficient optimizations for Rao-Blackwellized particle filters applied to solve the SLAM problem on grid maps. We are able to update the complex posterior requiring substantially less computational and memory re-

sources. This is achieved by performing the computations only for a set of representatives instead of for all particles. We extended a state-of-the-art mapping system in a way that the computation of the proposal distribution is significantly faster and needs only a fraction of the memory resources. The key idea is that clusters of particles can share large parts of their map model as well as an informed proposal distribution used to draw the next generation of particles.

With our optimizations, we are able to maintain more than one order of magnitude more samples and at the same time require less memory and computational resources compared to other state-of-the-art mapping techniques using Rao-Blackwellized particle filters. With this comparably high number of particles that we are able to maintain, we can compensate for the errors introduced by our approximations. Our approach has been implemented, tested, and evaluated based on real robots and standard log files used within the SLAM community to demonstrate the accuracy and benefits of our system.

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 FP6-004250-CoSy and FP6-IST-027140-BACS. The authors would like to acknowledge Mike Bosse and John Leonard for providing us the dataset of the MIT Killian Court, Patrick Beeson for the ACES dataset, and Dirk Hähnel for the Intel Research Lab.

References

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

[2] G. Grisetti, C. Stachniss, 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), Barcelona, Spain, 2005, pp. 2443–2448.

[3] J.-S. Gutmann, K. Konolige, Incremental mapping of large cyclic environments, in: Proc. of the IEEE Int. Symposium on Computational Intelligence in Robotics and Automation (CIRA), Monterey, CA, USA, 1999, pp. 318–325.

[4] D. Hähnel, W. Burgard, D. Fox, 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, pp. 206–211.

[5] M. Montemerlo, S. T. D. Koller, 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), Acapulco, Mexico, 2003, pp. 1151–1156.

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

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

[8] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, H. Durrant-Whyte, Simultaneous localization and mapping with sparse extended information filters, Int. Journal of Robotics Research 23 (7/8).

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

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

[11] R. Smith, M. Self, P. Cheeseman, Estimating uncertain spatial realtionships in robotics, in: I. Cox, G. Wilfong (Eds.), Autonomous Robot Vehicles, Springer Verlag, 1990, pp. 167–193.

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

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

[14] R. Eustice, M. Walter, J. Leonard, Sparse extended information filters: Insights into sparsification, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Edmonton, Cananda, 2005, pp. 641–648.

[15] M. Paskin, Thin junction tree filters for simultaneous localization and mapping, in: Proc. of the Int. Conf. on Artificial Intelligence (IJCAI), Acapulco, Mexico, 2003, pp. 1157–1164.

[16] F. Lu, E. Milios, Globally consistent range scan alignment for environment mapping, Journal of Autonomous Robots 4 (1997) 333–349.

[17] B. Lisien, D. S. D. Morales, G. Kantor, I. Rekleitis, H. Choset, Hierarchical simultaneous localization and mapping, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 2003, pp. 448–453.

[18] M. Bosse, P. Newman, J. Leonard, S. Teller, An ALTAS framework for scalable mapping, in: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA), Taipei, Taiwan, 2003, pp. 1899–1906.

[19] C. Estrada, J. Neira, J. Tardós, Hierachical slam: Real-time accurate mapping of large environments, IEEE Transactions on Robotics 21 (4) (2005) 588–596.

[20] A. Doucet, On sequential simulation-based methods for bayesian filtering, Tech. rep., Signal Processing Group, Dept. of Engeneering, University of Cambridge (1998).

[21] C. Stachniss, G. Grisetti, Mapping results obtained with Rao-Blackwellized particle filters, http://www.informatik.uni-freiburg.de/∼stachnis/research/rbpfmapper/ (2004).

[22] C. Stachniss, D. Hähnel, W. Burgard, G. Grisetti, On actively closing loops in grid-based fastslam, Advanced Robotics 19 (10) (2005) 1059–1080.