Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters

Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard

***Abstract*—Recently, Rao-Blackwellized particle filters (RBPF) have been introduced as an effective means to solve the simultaneous localization and mapping problem. This approach uses a particle filter in which each particle carries an individual map of the environment. Accordingly, a key question is how to reduce the number of particles. In this paper, we present adaptive techniques for reducing this number in a RBPF for learning grid maps. We propose an approach to compute an accurate proposal distribution, taking into account not only the movement of the robot, but also the most recent observation. This drastically decreases the uncertainty about the robot's pose in the prediction step of the filter. Furthermore, we present an approach to selectively carry out resampling operations, which seriously reduces the problem of particle depletion. Experimental results carried out with real mobile robots in large-scale indoor, as well as outdoor, environments illustrate the advantages of our methods over previous approaches.**

***Index Terms*—Adaptive resampling, improved proposal, motion model, Rao-Blackwellized particle filter (RBPF), simultaneous localization and mapping (SLAM).**

I. Introduction

BUILDING maps is one of the fundamental tasks of mobile robots. In the literature, the mobile-robot mapping problem is often referred to as the simultaneous localization and mapping (SLAM) problem [1]–[9]. It is considered to be a complex problem, because for localization, a robot needs a consistent map, and for acquiring a map, a robot requires a good estimate of its location. This mutual dependency between the pose and the map estimates makes the SLAM problem hard, and requires searching for a solution in a high-dimensional space.

Manuscript received July 18, 2005; revised June 30, 2006. This paper was recommended for publication by Associate Editor G. Sukhatme and Editor L. Parker upon evaluation of the reviewers' comments. This work was supported in part by the Marie Curie Program under Contract HPMT-CT-2001-00251, in part by the German Research Foundation (DFG) under Contract SFB/TR-8 (A3), and in part by the EC under Contracts FP6-004250-CoSy, FP6-IST-027140-BACS, and FP6-2005-IST-5-muFly. This paper was presented in part at the IEEE International Conference on Robotics and Automation, Barcelona, Spain, June 2005.

G. Grisetti is with the Department of Computer Science, University of Freiburg, D-79110 Freiburg, Germany, and also with the Dipartimento Informatica e Sistemistica, Universitá "La Sapienza," I-00198 Rome, Italy (e-mail: grisetti@informatik.uni-freiburg.de).

C. Stachniss is with the Department of Computer Science, University of Freiburg, D-79110 Freiburg, Germany, and also with the Eidgenössische Technische Hochschule Zurich (ETH), IRIS, ASL, 8092 Zurich, Switzerland (e-mail: stachnis@informatik.uni-freiburg.de).

W. Burgard is with the Department of Computer Science, University of Freiburg, D-79110 Freiburg, Germany (e-mail: burgard@informatik.uni-freiburg.de).

Color versions of Figs. 2–6, 8, 10, and 11 are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2006.889486

Murphy, Doucet, and colleagues [2], [8] introduced Rao-Blackwellized particle filters (RBPF) as an effective means to solve the SLAM problem. The main problem of the Rao-Blackwellized approaches is their complexity, measured in terms of the number of particles required to build an accurate map. Therefore, reducing this quantity is one of the major challenges for this family of algorithms. Additionally, the resampling step can potentially eliminate the correct particle. This effect is also known as the particle-depletion problem, or particle impoverishment [10].

In this paper, we present two approaches to substantially increase the performance of RBPFs applied to solve the SLAM problem with grid maps:

- a proposal distribution that considers the accuracy of the robot's sensors and allows us to draw particles in a highly accurate manner;
- an adaptive resampling technique which maintains a reasonable variety of particles, and in this way, enables the algorithm to learn an accurate map while reducing the risk of particle depletion.

The proposal distribution is computed by evaluating the likelihood around a particle-dependent most-likely pose, obtained by a scan-matching procedure combined with odometry information. In this way, the most recent sensor observation is taken into account for creating the next generation of particles. This allows us to estimate the state of the robot according to a more informed (and thus, more accurate) model than the one obtained based only on the odometry information. The use of this refined model has two effects. The map is more accurate, since the current observation is incorporated into the individual maps *after* considering its effect on the pose of the robot. This significantly reduces the estimation error, so that fewer particles are required to represent the posterior. The second approach, the adaptive resampling strategy, allows us to perform a resampling step only when needed, and in this way, keeps a reasonable particle diversity. This results in a significantly reduced risk of particle depletion.

The work presented in this paper is an extension of our previous work [11], as it further optimizes the proposal distribution to even more accurately draw the next generation of particles. Furthermore, we added a complexity analysis, a formal description of the techniques used, and provide more detailed experiments in this paper. Our approach has been validated by a set of systematic experiments in large-scale indoor and outdoor environments. In all experiments, our approach generated highly accurate metric maps. Additionally, the number of the required particles is one order of magnitude lower than with previous approaches.

This paper is organized as follows. After explaining how a Rao-Blackwellized filter can be used in general to solve the SLAM problem, we describe our approach in Section III. We then provide implementation details in Section IV. Experiments carried out on real robots are presented in Section VI. Finally, Section VII discusses related approaches.

II. Mapping With RBPFs

According to Murphy [8], the key idea of the RBPF for SLAM is to estimate the joint posterior $p(x_{1:t}, m \mid z_{1:t}, u_{1:t-1})$ about the map m and the trajectory $x_{1:t} = x_1, \ldots, x_t$ of the robot. This estimation is performed given the observations $z_{1:t} = z_1, \ldots, z_t$ and the odometry measurements $u_{1:t-1} = u_1, \ldots, u_{t-1}$ obtained by the mobile robot. The RBPF for SLAM makes use of the following factorization:

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

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

Typically, (1) can be calculated efficiently, since the posterior over maps $p(m \mid x_{1:t}, z_{1:t})$ can be computed analytically using "mapping with known poses" [12], since $x_{1:t}$ and $z_{1:t}$ are known.

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

One of the most common particle-filtering algorithms is the sampling importance resampling (SIR) filter. A Rao-Blackwellized SIR filter for mapping incrementally processes the sensor observations and the odometry readings as they are available. It updates the set of samples that represents the posterior about the map and the trajectory of the vehicle. The process can be summarized by the following four steps.

1) *Sampling*: The next generation of particles $\{x_t^{(i)}\}$ is obtained from the generation $\{x_{t-1}^{(i)}\}$ by sampling from the proposal distribution π. Often, a probabilistic odometry motion model is used as the proposal distribution.
2) *Importance Weighting*: An individual importance weight $w_t^{(i)}$ is assigned to each particle, according to the importance sampling principle

$$w_t^{(i)} = \frac{p(x_{1:t}^{(i)} \mid z_{1:t}, u_{1:t-1})}{\pi(x_{1:t}^{(i)} \mid z_{1:t}, u_{1:t-1})}. \quad (2)$$

The weights account for the fact that the proposal distribution π is, in general, not equal to the target distribution of successor states.

3) *Resampling*: Particles are drawn with replacement proportional to their importance weight. This step is necessary, since only a finite number of particles is used to approximate a continuous distribution. Furthermore, resampling allows us to apply a particle filter in situations in which the target distribution differs from the proposal. After resampling, all the particles have the same weight.
4) *Map Estimation*: For each particle, the corresponding map estimate $p(m^{(i)} \mid x_{1:t}^{(i)}, z_{1:t})$ is computed, based on the trajectory $x_{1:t}^{(i)}$ of that sample and the history of observations $z_{1:t}$.

The implementation of this schema requires evaluating the weights of the trajectories from scratch whenever a new observation is available. Since the length of the trajectory increases over time, this procedure would lead to an obviously inefficient algorithm. According to Doucet *et al.* [13], we obtain a recursive formulation to compute the importance weights by restricting the proposal π to fulfill the following assumption:

$$\pi(x_{1:t} \mid z_{1:t}, u_{1:t-1}) = \pi(x_t \mid x_{1:t-1}, z_{1:t}, u_{1:t-1}) \cdot \pi(x_{1:t-1} \mid z_{1:t-1}, u_{1:t-2}). \quad (3)$$

Based on (2) and (3), the weights are computed as

$$w_t^{(i)} = \frac{p(x_{1:t}^{(i)} \mid z_{1:t}, u_{1:t-1})}{\pi(x_{1:t}^{(i)} \mid z_{1:t}, u_{1:t-1})} \quad (4)$$

$$= \frac{\eta p(z_t \mid x_{1:t}^{(i)}, z_{1:t-1}) p(x_t^{(i)} \mid x_{t-1}^{(i)}, u_{t-1})}{\pi(x_t^{(i)} \mid x_{1:t-1}^{(i)}, z_{1:t}, u_{1:t-1})} \cdot \underbrace{\frac{p(x_{1:t-1}^{(i)} \mid z_{1:t-1}, u_{1:t-2})}{\pi(x_{1:t-1}^{(i)} \mid z_{1:t-1}, u_{1:t-2})}}_{w_{t-1}^{(i)}} \quad (5)$$

$$\propto \frac{p(z_t \mid m_{t-1}^{(i)}, x_t^{(i)}) p(x_t^{(i)} \mid x_{t-1}^{(i)}, u_{t-1})}{\pi(x_t \mid x_{1:t-1}^{(i)}, z_{1:t}, u_{1:t-1})} \cdot w_{t-1}^{(i)}. \quad (6)$$

Here $\eta = 1/p(z_t \mid z_{1:t-1}, u_{1:t-1})$ is a normalization factor resulting from Bayes' rule that is equal for all particles.

Most of the existing particle filter applications rely on the recursive structure of (6). Whereas the generic algorithm specifies a framework that can be used for learning maps, it leaves open how the proposal distribution should be computed and when the resampling step should be carried out. Throughout the remainder of this paper, we describe a technique that computes an accurate proposal distribution, and that adaptively performs resampling.

III. RBPF With Improved Proposals and Adaptive Resampling

In the literature, several methods for computing improved proposal distributions and for reducing the risk of particle depletion have been proposed [13]–[15]. Our approach applies two concepts that have previously been identified as key prerequisites for efficient particle-filter implementations (see Doucet *et al.* [13]), namely, the computation of an improved proposal distribution and an adaptive resampling technique.

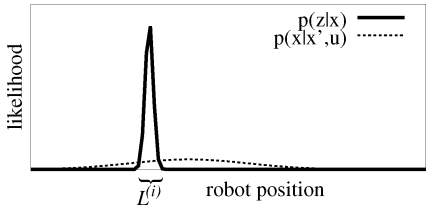

Fig. 1. The two components of the motion model. Within the interval $L^{(i)}$, the product of both functions is dominated by the observation likelihood in case an accurate sensor is used.

A. On the Improved Proposal Distribution

As described in Section II, one needs to draw samples from a proposal distribution π in the prediction step in order to obtain the next generation of particles. Intuitively, the better the proposal distribution approximates the target distribution, the better the performance of the filter. For instance, if we were able to directly draw samples from the target distribution, the importance weights would become equal for all particles, and the resampling step would no longer be needed. Unfortunately, in the context of SLAM, a closed form of this posterior is not available, in general.

As a result, typical particle filter applications [16], [7] use the odometry motion model as the proposal distribution. This motion model has the advantage that it is easy to compute for most types of robots. Furthermore, the importance weights are then computed according to the observation model $p(z_t \mid m, x_t)$. This becomes clear by replacing π in (6) by the motion model $p(x_t \mid x_{t-1}, u_{t-1})$

$$w_t^{(i)} = w_{t-1}^{(i)} \frac{\eta p(z_t \mid m_{t-1}^{(i)}, x_t^{(i)}) p(x_t^{(i)} \mid x_{t-1}^{(i)}, u_{t-1})}{p(x_t^{(i)} \mid x_{t-1}^{(i)}, u_{t-1})} \tag{7}$$

$$\propto w_{t-1}^{(i)} \cdot p(z_t | m_{t-1}^{(i)}, x_t^{(i)}). \tag{8}$$

This proposal distribution, however, is suboptimal, especially when the sensor information is *significantly more precise* than the motion estimate of the robot based on the odometry, which is typically the case if a robot equipped with a laser range finder [e.g., with a SICK Laser Measurement Sensor (LMS)]. Fig. 1 illustrates a situation in which the meaningful area of the observation likelihood is substantially smaller than the meaningful area of the motion model. When using the odometry model as the proposal distribution in such a case, the importance weights of the individual samples can differ significantly from each other, since only a fraction of the drawn samples cover the regions of state space that have a high likelihood under the observation model (area $L^{(i)}$ in Fig. 1). As a result, one needs a comparably high number of samples to sufficiently cover the regions with high observation likelihood.

A common approach, especially in localization, is to use a smoothed likelihood function, which avoids particles close to the meaningful area getting a too-low importance weight. However, this approach discards useful information gathered by the sensor and, at least in our experience, often leads to less accurate maps in the SLAM context.

To overcome this problem, one can consider the most recent sensor observation z_t when generating the next generation of samples. By integrating z_t into the proposal, one can focus the sampling on the meaningful regions of the observation likelihood. According to Doucet [17], the distribution

$$p(x_t \mid m_{t-1}^{(i)}, x_{t-1}^{(i)}, z_t, u_{t-1}) = \frac{p(z_t \mid m_{t-1}^{(i)}, x_t) p(x_t \mid x_{t-1}^{(i)}, u_{t-1})}{p(z_t \mid m_{t-1}^{(i)}, x_{t-1}^{(i)}, u_{t-1})} \tag{9}$$

is the optimal proposal distribution with respect to the variance of the particle weights. Using that proposal, the computation of the weights turns into

$$w_t^{(i)} = w_{t-1}^{(i)} \frac{\eta p(z_t \mid m_{t-1}^{(i)}, x_t^{(i)}) p(x_t^{(i)} \mid x_{t-1}^{(i)}, u_{t-1})}{p(x_t \mid m_{t-1}^{(i)}, x_{t-1}^{(i)}, z_t, u_{t-1})} \tag{10}$$

$$\propto w_{t-1}^{(i)} \frac{p(z_t \mid m_{t-1}^{(i)}, x_t^{(i)}) p(x_t^{(i)} \mid x_{t-1}^{(i)}, u_{t-1})}{\frac{p(z_t | m_{t-1}^{(i)}, x_t) p(x_t | x_{t-1}^{(i)}, u_{t-1})}{p(z_t | m_{t-1}^{(i)}, x_{t-1}^{(i)}, u_{t-1})}} \tag{11}$$

$$= w_{t-1}^{(i)} \cdot p(z_t \mid m_{t-1}^{(i)}, x_{t-1}^{(i)}, u_{t-1}) \tag{12}$$

$$= w_{t-1}^{(i)} \cdot \int p(z_t \mid x') p(x' \mid x_{t-1}^{(i)}, u_{t-1}) \, dx'. \tag{13}$$

When modeling a mobile robot equipped with an accurate sensor like, e.g., a laser range finder, it is convenient to use such an improved proposal, since the accuracy of the laser range finder leads to extremely peaked likelihood functions. In the context of landmark-based SLAM, Montemerlo *et al.* [6] presented an RBPF that uses a Gaussian approximation of the improved proposal. This Gaussian is computed for each particle, using a Kalman filter that estimates the pose of the robot. This approach can be used when the map is represented by a set of features, and if the error affecting the feature detection is assumed to be Gaussian. In this paper, we transfer the idea of computing an improved proposal to the situation in which dense grid maps are used, instead of landmark-based representations.

B. Efficient Computation of the Improved Proposal

When modeling the environment with grid maps, a closed-form approximation of an informed proposal is not directly available, due to the unpredictable shape of the observation likelihood function.

In theory, an approximated form of the informed proposal can be obtained using the *adapted* particle filter [15]. In this framework, the proposal for each particle is constructed by computing a sampled estimate of the optimal proposal given in (9). In the SLAM context, one would first have to sample a set of potential poses x_j of the robot from the motion model $p(x_t \mid x_{t-1}^{(i)}, u_{t-1})$. In a second step, these samples need to be weighed by the observation likelihood to obtain an approximation of the optimal proposal. However, if the observation likelihood is peaked, the number of pose samples x_j that has to be sampled from the motion model is high, since a dense sampling is needed for sufficiently capturing the typically small areas of high likelihood. This results in a similar problem to using the motion model as the proposal: a high number of samples is needed to sufficiently cover the meaningful region of the distribution.

One of our observations is that in the majority of cases, the target distribution has only a limited number of maxima, and it mostly has only a single one. This allows us to sample positions x_j covering only the area surrounding these maxima. Ignoring

the less meaningful regions of the distribution saves a significant amount of computational resources, since it requires fewer samples. In the previous version of this paper [11], we approximated $p(x_t \mid x_{t-1}^{(i)}, u_{t-1})$ by a constant k within the interval $L^{(i)}$ (see also Fig. 1), given by

$$L^{(i)} = \left\{ x \;\middle|\; p(z_t \mid m_{t-1}^{(i)}, x) > \epsilon \right\}. \tag{14}$$

In our current approach, we consider both components, the observation likelihood and the motion model within that interval $L^{(i)}$. We locally approximate the posterior $p(x_t \mid m_{t-1}^{(i)}, x_{t-1}^{(i)}, z_t, u_{t-1})$ around the maximum of the likelihood function reported by a scan registration procedure.

To efficiently draw the next generation of samples, we compute a Gaussian approximation $\mathcal{N}$ based on that data. The main differences from previous approaches is that we first use a scan-matcher to determine the meaningful area of the observation likelihood function. We then sample in that meaningful area and evaluate the sampled points based on the target distribution. For each particle i, the parameters $\mu_t^{(i)}$ and $\Sigma_t^{(i)}$ are determined individually for K sampled points $\{x_j\}$ in the interval $L^{(i)}$. Furthermore, we take into account the odometry information when computing the mean $\mu^{(i)}$ and the variance $\Sigma^{(i)}$. We estimate the Gaussian parameters as

$$\mu_t^{(i)} = \frac{1}{\eta^{(i)}} \cdot \sum_{j=1}^{K} x_j \cdot p(z_t \mid m_{t-1}^{(i)}, x_j) \cdot p(x_j \mid x_{t-1}^{(i)}, u_{t-1}) \tag{15}$$

$$\Sigma_t^{(i)} = \frac{1}{\eta^{(i)}} \cdot \sum_{j=1}^{K} p(z_t \mid m_{t-1}^{(i)}, x_j) \cdot p(x_j \mid x_{t-1}^{(i)}, u_{t-1}) \cdot (x_j - \mu_t^{(i)})(x_j - \mu_t^{(i)})^T \tag{16}$$

with the normalization factor

$$\eta^{(i)} = \sum_{j=1}^{K} p(z_t \mid m_{t-1}^{(i)}, x_j) \cdot p(x_j \mid x_{t-1}^{(i)}, u_{t-1}). \tag{17}$$

In this way, we obtain a closed-form approximation of the optimal proposal which enables us to efficiently obtain the next generation of particles. Using this proposal distribution, the weights can be computed as

$$\begin{aligned} w_t^{(i)} &= w_{t-1}^{(i)} \cdot p(z_t \mid m_{t-1}^{(i)}, x_{t-1}^{(i)}, u_{t-1}) && (18)\\ &= w_{t-1}^{(i)} \cdot \int p(z_t \mid m_{t-1}^{(i)}, x') \cdot p(x' \mid x_{t-1}^{(i)}, u_{t-1})\, dx \\ &\simeq w_{t-1}^{(i)} \cdot \sum_{j=1}^{K} p(z_t \mid m_{t-1}^{(i)}, x_j) \cdot p(x_j \mid x_{t-1}^{(i)}, u_{t-1}) \\ &= w_{t-1}^{(i)} \cdot \eta^{(i)}. && (19) \end{aligned}$$

Note that $\eta^{(i)}$ is the same normalization factor that is used in the computation of the Gaussian approximation of the proposal in (17).

C. Discussion About the Improved Proposal

The computations presented in this section enable us to determine the parameters of a Gaussian proposal distribution for each particle individually. The proposal takes into account the most recent odometry reading and laser observation, while at the same time allowing us efficient sampling. The resulting densities have a much lower uncertainty, compared with situations in which the odometry motion model is used. To illustrate this fact, Fig. 2 depicts typical particle distributions obtained with our approach. In case of a straight featureless corridor, the samples are typically spread along the main direction of the corridor, as depicted in Fig. 2(a). Fig. 2(b) illustrates the robot reaching the end of such a corridor. As can be seen, the uncertainty in the direction of the corridor decreases, and all samples are centered around a single point. In contrast to that, Fig. 2(c) shows the resulting distribution when sampling from the raw motion model.

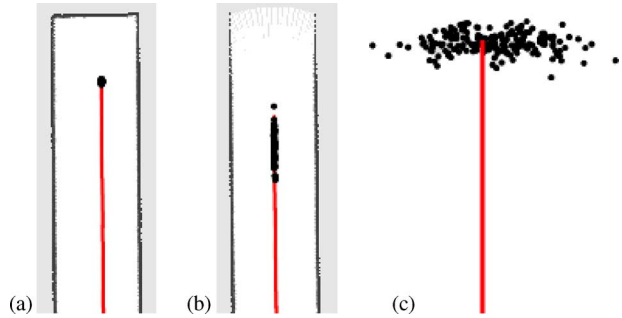

Fig. 2. Particle distributions typically observed during mapping. In an open corridor, the particles distribute along the corridor (a). In a dead-end corridor, the uncertainty is small in all dimensions (b). Such posteriors are obtained because we explicitly take into account the most recent observation when sampling the next generation of particles. In contrast to that, the raw odometry motion model leads less peaked posteriors (c).

As explained above, we use a scan-matcher to determine the mode of the meaningful area of the observation likelihood function. In this way, we focus the sampling on the important regions. Most existing scan-matching algorithms maximize the observation likelihood, given a map and an initial guess of the robot's pose. When the likelihood function is multimodal, which can occur when, e.g., closing a loop, the scan-matcher returns for each particle the maximum which is closest to the initial guess. In general, it can happen that additional maxima in the likelihood function are missed, since only a single mode is reported. However, since we perform frequent filter updates (after each movement of 0.5 m or a rotation of 25°) and limit the search area of the scan-matcher, we consider that the distribution has only a single mode when sampling data points to compute the Gaussian proposal. Note that in situations like a loop closure, the filter is still able to keep multiple hypotheses, because the initial guess for the starting position of the scan-matcher when reentering a loop is different for each particle.

Nevertheless, there are situations in which the filter can, at least in theory, become overly confident. This might happen in extremely cluttered environments and when the odometry is highly affected by noise. A solution to this problem is to track the multiple modes of the scan-matcher and repeat the sampling process separately for each node. However, in our experiments, carried out using real robots, we never encountered such a situation.

During filtering, it can happen that the scan-matching process fails because of poor observations or a too-small overlapping area between the current scan and the previously computed map.

In this case, the raw motion model of the robot which is illustrated in Fig. 2(c) is used as a proposal. Note that such situations occur rarely in real datasets (see also Section VI-E).

D. Adaptive Resampling

A further aspect that has a major influence on the performance of a particle filter is the resampling step. During resampling, particles with a low importance weight $w^{(i)}$ are typically replaced by samples with a high weight. On the one hand, resampling is necessary, since only a finite number of particles are used to approximate the target distribution. On the other hand, the resampling step can remove good samples from the filter, which can lead to particle impoverishment. Accordingly, it is important to find a criterion for deciding when to perform the resampling step. Liu [18] introduced the so-called effective sample size to estimate how well the current particle set represents the target posterior. In this paper, we compute this quantity according to the formulation of Doucet *et al.* [13] as

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^{N} \left(\tilde{w}^{(i)}\right)^2} \tag{20}$$

where $\tilde{w}^{(i)}$ refers to the normalized weight of particle i.

The intuition behind N_{eff} is as follows. If the samples were drawn from the target distribution, their importance weights would be equal to each other, due to the importance-sampling principle. The worse the approximation of the target distribution, the higher the variance of the importance weights. Since N_{eff} can be regarded as a measure of the dispersion of the importance weights, it is a useful measure to evaluate how well the particle set approximates the target posterior. Our algorithm follows the approach proposed by Doucet *et al.* [13] to determine whether or not the resampling step should be carried out. We resample each time N_{eff} drops below the threshold of $N/2$, where N is the number of particles. In extensive experiments, we found that this approach drastically reduces the risk of replacing good particles, because the number of resampling operations is reduced, and they are only performed when needed.

Algorithm

The overall process is summarized in **Algorithm 1**. Each time a new measurement tuple (u_{t-1}, z_t) is available, the proposal is computed for each particle individually, and is then used to update that particle. This results in the following steps.

1) An initial guess $x_t'^{(i)} = x_{t-1}^{(i)} \oplus u_{t-1}$ for the robot's pose, represented by the particle i, is obtained from the previous pose $x_{t-1}^{(i)}$ of that particle and the odometry measurements u_{t-1} collected since the last filter update. Here, the operator $\oplus$ corresponds to the standard pose compounding operator [19].
2) A scan-matching algorithm is executed based on the map $m_{t-1}^{(i)}$, starting from the initial guess $x_t'^{(i)}$. The search performed by the scan-matcher is bounded to a limited region around $x_t'^{(i)}$. If the scan-matching reports a failure, the pose and the weights are computed according to the motion model (and steps 3) and 4) are ignored).
3) A set of sampling points is selected in an interval around the pose $\hat{x}_t^{(i)}$ reported scan-matcher. Based on these points, the mean and the covariance matrix of the proposal are computed by pointwise evaluating the target distribution $p(z_t \mid m_{t-1}^{(i)}, x_j)p(x_j \mid x_{t-1}^{(i)}, u_{t-1})$ in the sampled positions x_j. During this phase, the weighting factor $\eta^{(i)}$ is also computed, according to (17).
4) The new pose $x_t^{(i)}$ of the particle i is drawn from the Gaussian approximation $\mathcal{N}(\mu_t^{(i)}, \Sigma_t^{(i)})$ of the improved proposal distribution.
5) Update of the importance weights.
6) The map $m^{(i)}$ of particle i is updated according to the drawn pose $x_t^{(i)}$ and the observation z_t.

After computing the next generation of samples, a resampling step is carried out, depending on the value of N_{eff}.

Algorithm 1 Improved RBPF for Map Learning

Require:

$\mathcal{S}_{t-1}$, the sample set of the previous time step

z_t, the most recent laser scan

u_{t-1}, the most recent odometry measurement

Ensure:

$\mathcal{S}_t$, the new sample set

$\mathcal{S}_t = \{\}$

for all $s_{t-1}^{(i)} \in \mathcal{S}_{t-1}$ **do**

$\langle x_{t-1}^{(i)}, w_{t-1}^{(i)}, m_{t-1}^{(i)} \rangle = s_{t-1}^{(i)}$

// scan-matching

$x_t'^{(i)} = x_{t-1}^{(i)} \oplus u_{t-1}$

$\hat{x}_t^{(i)} = \text{argmax}_x \, p(x \mid m_{t-1}^{(i)}, z_t, x_t'^{(i)})$

if $\hat{x}_t^{(i)} = \textbf{failure}$ **then**

$x_t^{(i)} \sim p(x_t \mid x_{t-1}^{(i)}, u_{t-1})$

$w_t^{(i)} = w_{t-1}^{(i)} \cdot p(z_t \mid m_{t-1}^{(i)}, x_t^{(i)})$

else

// sample around the mode

for $k = 1, \ldots, K$ **do**

$x_k \sim \{x_j | \, |x_j - \hat{x}^{(i)}| < \Delta\}$

end for

// compute Gaussian proposal

$\mu_t^{(i)} = (0, 0, 0)^T$

$\eta^{(i)} = 0$

for all $x_j \in \{x_1, \ldots, x_K\}$ **do**

$\mu_t^{(i)} = \mu_t^{(i)} + x_j \cdot p(z_t \mid m_{t-1}^{(i)}, x_j) \cdot p(x_t \mid x_{t-1}^{(i)}, u_{t-1})$

$\eta^{(i)} = \eta^{(i)} + p(z_t \mid m_{t-1}^{(i)}, x_j) \cdot p(x_t \mid x_{t-1}^{(i)}, u_{t-1})$

end for

$\mu_t^{(i)} = \mu_t^{(i)}/\eta^{(i)}$

$\Sigma_t^{(i)} = \mathbf{0}$

for all $x_j \in \{x_1, \ldots, x_K\}$ **do**

$$\Sigma_t^{(i)} = \Sigma_t^{(i)} + (x_j - \mu^{(i)})(x_j - \mu^{(i)})^T \cdot p(z_t \mid m_{t-1}^{(i)}, x_j) \cdot p(x_j \mid x_{t-1}^{(i)}, u_{t-1})$$

end for

$\Sigma_t^{(i)} = \Sigma_t^{(i)}/\eta^{(i)}$

// sample new pose

$x_t^{(i)} \sim \mathcal{N}(\mu_t^{(i)}, \Sigma_t^{(i)})$

// update importance weights

$w_t^{(i)} = w_{t-1}^{(i)} \cdot \eta^{(i)}$

end if

// update map

$m_t^{(i)} = \text{integrateScan}(m_{t-1}^{(i)}, x_t^{(i)}, z_t)$

// update sample set

$\mathcal{S}_t = \mathcal{S}_t \cup \{\langle x_t^{(i)}, w_t^{(i)}, m_t^{(i)}\rangle\}$

end for

$N_{\text{eff}} = 1/\sum_{i=1}^{N} \left(\tilde{w}^{(i)}\right)^2$

if $N_{\text{eff}} < T$ **then**

$\mathcal{S}_t = \text{resample}\ (\mathcal{S}_t)$

end if

IV. Implementation Issues

This section provides additional information about implementation details used in our current system. These issues are not required for the understanding of the general approach, but complete the precise description of our mapping system. In the following, we briefly explain the used scan-matching approach, the observation model, and how to pointwise evaluate the motion model.

Our approach applies a scan-matching technique on a per-particle basis. In general, an arbitrary scan-matching technique can be used. In our implementation, we use the scan-matcher "vasco," which is part of the Carnegie Mellon Robot Navigation Toolkit (CARMEN) [20], [21]. This scan-matcher aims to find the most likely pose by matching the current observation against the map constructed so far

$$\hat{x}_t^{(i)} = \underset{x}{\text{argmax}}\, p(x \mid m_{t-1}^{(i)}, z_t, x_t'^{(i)}) \tag{21}$$

where $x_t'^{(i)}$ is the initial guess. The scan-matching technique performs a gradient-descent search on the likelihood function of the current observation, given the grid map. Note that in our mapping approach, the scan-matcher is only used for finding the local maximum of the observation likelihood function. In practice, any scan-matching technique which is able to compute the best alignment between a reference map $m_{t-1}^{(i)}$ and the current scan z_t given an initial guess $x_t'^{(i)}$ can be used.

In order to solve (21), one applies Bayes' rule and seeks for the pose with the highest observation likelihood $p(z_t \mid m, x)$. To compute the likelihood of an observation, we use the so called "beam-endpoint model" [22]. In this model, the individual beams within a scan are considered to be independent. Furthermore, the likelihood of a beam is computed based on the distance between the endpoint of the beam and the closest obstacle from that point. To achieve a fast computation, one typically uses a convolved local grid map.

Additionally, the construction of our proposal requires evaluating $p(z_t \mid m_{t-1}^{(i)}, x_j)p(x_j \mid x_{t-1}^{(i)}, u_{t-1})$ at the sampled points x_j. We compute the first component according to the previously mentioned beam-endpoint model. To evaluate the second term, several closed-form solutions for the motion estimate are available. The different approaches mainly differ in the way the kinematics of the robot are modeled. In our current implementation, we compute $p(x_j \mid x_{t-1}, u_{t-1})$ according to the Gaussian approximation of the odometry motion model described in [23]. We obtain this approximation through Taylor expansion in an extended Kalman filter (EKF)-style procedure. In general, there are more sophisticated techniques estimating the motion of the robot. However, we use that model to estimate a movement between two filter updates, which is performed after the robot traveled around 0.5 m. In this case, this approximation works well, and we did not observe a significant difference between the EKF-like model and the, in general, more accurate sample-based velocity motion model [23].

V. Complexity

This section discusses the complexity of the presented approach to learn grid maps with an RBPF. Since our approach uses a sample set to represent the posterior about maps and poses, the number N of samples is the central quantity. To compute the proposal distribution, our approach samples around the most likely position reported by the scan-matcher. This sampling is done for each particle a constant number of times (K), and there is no dependency between the particles when computing the proposal. Furthermore, the most recent observation z_t used to compute $\mu^{(i)}$ and $\Sigma^{(i)}$ covers only an area of the map m (bounded by the odometry error and the maximum range of the sensor), so the complexity depends only on the number N of particles. The same holds for the update of the individual maps associated with each of the particles.

During the resampling step, the information associated with a particle needs to be copied. In the worst case, $N-1$ samples are replaced by a single particle. In our current system, each particle stores and maintains its own grid map. To duplicate a particle, we therefore have to copy the whole map. As a result, a resampling action introduces a worst-case complexity of $O(NM)$, where M is the size of the corresponding grid map. However, using the adaptive resampling technique, only a very few resampling steps are required during mapping.

TABLE I
COMPLEXITY OF THE DIFFERENT OPERATIONS FOR INTEGRATING ONE OBSERVATION

Operation	Complexity
Computation of the proposal distribution	$O(N)$
Update of the grid map	$O(N)$
Computation of the weights	$O(N)$
Test if resampling is required	$O(N)$
Resampling	$O(NM)$

To decide whether or not a resampling is needed, the effective sample size [see (20)] needs to be taken into account. Again, the computation of the quantity introduces a complexity of $O(N)$.

As a result, if no resampling operation is required, the overall complexity for integrating a single observation depends only linearly on the number of particles. If a resampling is required, the additional factor M which represents the size of the map is introduced, and leads to a complexity of $O(NM)$. The complexity of each individual operation is depicted in Table I.

Note that the complexity of the resampling step can be reduced by using a more intelligent map representation, as done in distributed particle (DP)-SLAM [3], for example. It can be shown that in this case, the complexity of a resampling step is reduced to $O(AN^2 \log N)$, where A is the area covered by the sensor. However, building an improved map representation is not the aim of this paper. We actually see our approach as orthogonal to DP-SLAM, because both techniques can be combined. Furthermore, in our experiments using real-world data sets, we figured out the resampling steps are not the dominant part, and they occur rarely, due to the adaptive-resampling strategy.

Fig. 3. Different types of robot used to acquire real robot data used for mapping (ActivMedia Pioneer 2 AT, Pioneer 2 DX-8, and an iRobot B21r).

Fig. 4. Intel Research Lab. The robot starts in the upper part of the circular corridor, and runs several times around the loop, before entering the rooms. The left image depicts the resulting map generated with 15 particles. The right image shows a cutout with 1-cm grid resolution to illustrate the accuracy of the map in the loop closure point.

VI. EXPERIMENTS

The approach described above has been implemented and tested using real robots and datasets gathered with real robots. Our mapping approach runs online on several platforms like ActivMedia Pioneer2 AT, Pioneer 2 DX-8, and iRobot B21r robots equipped with SICK LMS and PLS laser range finders (see Fig. 3). The experiments have been carried out in a variety of environments, and showed the effectiveness of our approach in indoor and outdoor settings. Most of the maps generated by our approach can be magnified up to a resolution of 1 cm, without observing considerable inconsistencies. Even in big real-world datasets covering an area of approximately 250 m by 250 m, our approach never required more than 80 particles to build accurate maps. In the reminder of this section, we discuss the behavior of the filter in different datasets. Furthermore, we give a quantitative analysis of the performance of the presented approach. Highly accurate grid maps have been generated with our approach from several datasets. These maps, raw data files, and an efficient implementation of our mapping system are available on the web [24].

A. Mapping Results

The datasets discussed here have been recorded at the Intel Research Lab in Seattle, WA, at the campus of the University of Freiburg, and at the Killian Court at MIT (Cambridge, MA). The maps of these environments are depicted in Figs. 4–6.

1) Intel Research Lab: The Intel Research Lab is depicted in the left image of Fig. 4 and has a size of 28 m by 28 m. The dataset has been recorded with a Pioneer II robot equipped with a SICK sensor. To successfully correct this dataset, our algorithm needed only 15 particles. As can be seen in the right image of Fig. 4, the quality of the final map is so high that the map can be magnified up to 1-cm resolution without showing any significant errors.

2) Freiburg Campus: The second dataset has been recorded outdoors at the Freiburg campus. Our system needed only 30 particles to produce a good quality map, such as the one shown in Fig. 5. Note that this environment partly violates the assumptions that the environment is planar. Additionally, there were objects like bushes and grass, as well as moving objects like cars and people. Despite the resulting spurious measurements, our algorithm was able to generate an accurate map.

3) MIT Killian Court: The third experiment was performed with a dataset acquired at the MIT Killian court[1] and the resulting map is depicted in Fig. 6. This dataset is extremely challenging, since it contains several nested loops, which can cause

[1]Note that there exist two different versions of that dataset on the web. One has a precorrected odometry and the other one has not. We used the raw version without precorrected odometry information.

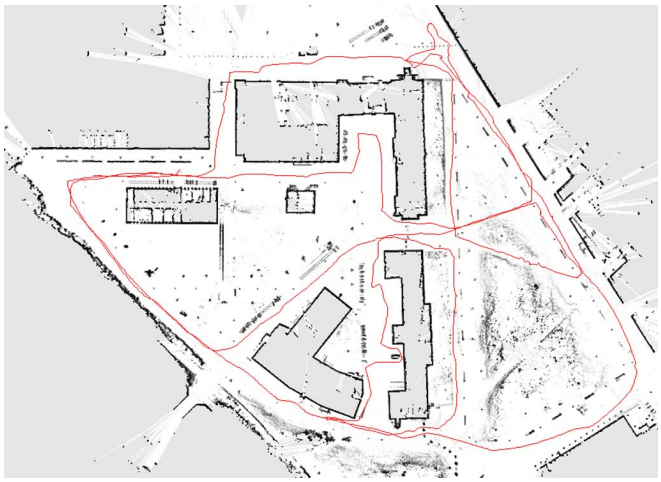

Fig. 5. Freiburg Campus. The robot first runs around the external perimeter in order to close the outer loop. Afterwards, the internal parts of the campus are visited. The overall trajectory has a length of 1.75 km and covers an area of approximately 250 m by 250 m. The depicted map was generated using 30 particles.

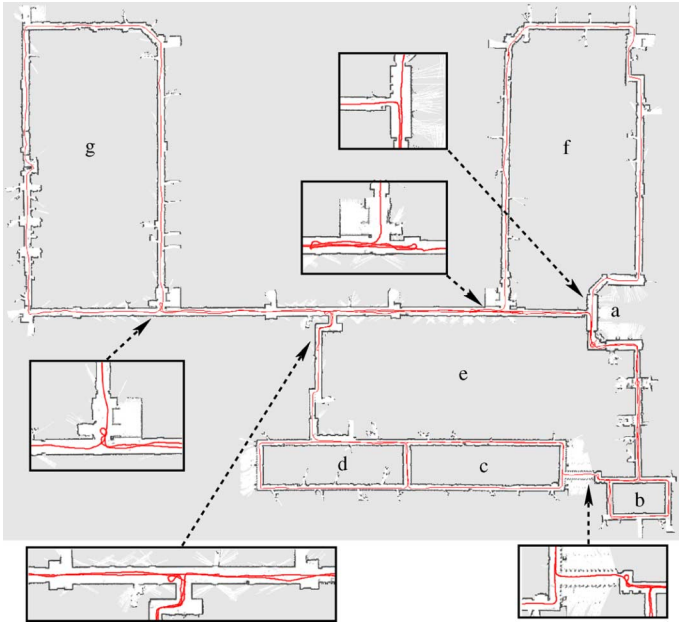

Fig. 6. MIT Killian Court. The robot starts from the point labeled *a* and then traverses the first loop labeled *b*. It then moves through the loops labeled *c* and *d*, and moves back to the place labeled *a* and the loop *b*. It then visits the two big loops labeled *f* and *g*. The environment has a size of 250 m by 215 m and the robot traveled 1.9 km. The depicted map has been generated with 80 particles. The rectangles show magnifications of several parts of the map.

an RBPF to fail due to particle depletion. Using this dataset, the selective resampling procedure turned out to be important. A consistent and topologically correct map can be generated with 60 particles. However, the resulting maps sometimes show artificial double walls. By employing 80 particles, it is possible to achieve high-quality maps.

B. Quantitative Results

In order to measure the improvement in terms of the number of particles, we compared the performance of our system using the informed proposal distribution with the approach done by Hähnel *et al.* [5]. Table II summarizes the number of particles needed by an RBPF for providing a topologically correct map in at least 60% of all applications of our algorithm.

TABLE II
NUMBER OF PARTICLES NEEDED BY OUR ALGORITHM COMPARED WITH THE APPROACH OF HÄHNEL [5]

Proposal Distribution	Intel	MIT	Freiburg
our approach	8	60	20
approach of [5]	40	400	400

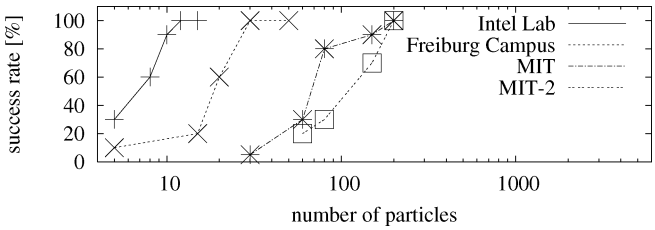

Fig. 7. Success rate of our algorithm in different environments, depending on the number of particles. Each success rate was determined using 20 runs. For the experiment MIT-2, we disabled the adaptive resampling.

It turns out that in all of the cases, the number of particles required by our approach was approximately one order of magnitude smaller than the one required by the other approach. Moreover, the resulting maps are better, due to our improved sampling process that takes the last reading into account.

Fig. 7 summarizes results about the success ratio of our algorithm in the environments considered here. The plots show the percentage of correctly generated maps, depending on the number of particles used. The question of whether a map is consistent or not has been evaluated by visual inspection in a blind fashion (the inspectors were not the authors). As a measure of success, we used the topological correctness.

C. Effects of Improved Proposals and Adaptive Resampling

The increased performance of our approach is due to the interplay of two factors, namely, the improved proposal distribution, which allows us to generate samples with a high likelihood, and the adaptive resampling controlled by monitoring N_{eff}. For proposals that do not consider the whole input history, it has been proven that N_{eff} can only decrease (stochastically) over time [13]. Only after a resampling operation, N_{eff} recovers its maximum value. It is important to notice that the behavior of N_{eff} depends on the proposal: the worse the proposal, the faster N_{eff} drops.

We found that the evolution of N_{eff} using our proposal distribution shows three different behaviors, depending on the information obtained from the robot's sensors. Fig. 8 illustrates the evolution of N_{eff} during an experiment. Whenever the robot moves through unknown terrain, N_{eff} typically drops slowly. This is because the proposal distribution becomes less peaked, and the likelihoods of observations often differ slightly. The second behavior can be observed when the robot moves through a known area. In this case, each particle keeps localized within its own map, due to the improved proposal distribution, and the weights are more or less equal. This results in a more or less constant behavior of N_{eff}. Finally, when closing a loop, some

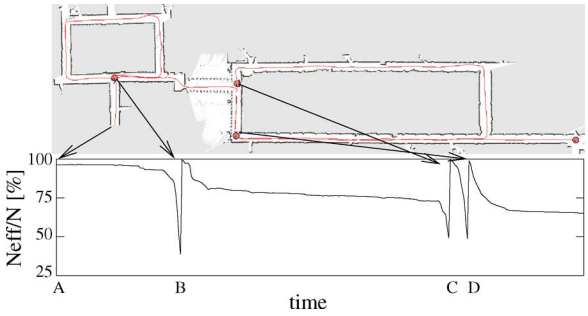

Fig. 8. Graph plots the evolution of the N_{eff} function over time during an experiment in the environment shown in the top image. At time B, the robot closes the small loop. At times C and D, resampling actions are carried after the robot closes the big loop.

particles are correctly aligned with their map, while others are not. The correct particles have a high weight, while the wrong ones have a low weight. Thus, the variance of the importance weights increases, and N_{eff} substantially drops.

Accordingly, the threshold criterion applied on N_{eff} typically forces a resampling action when the robot is closing a loop. In all other cases, the resampling is avoided, and in this way, the filter keeps a variety of samples in the particle set. As a result, the risk of the particle-depletion problem is seriously reduced. To analyze this, we performed an experiment in which we compared the success rate of our algorithm with that of a particle filter which resamples at every step. As Fig. 7 illustrates, our approach more often converged to the correct solution (MIT curve) for the MIT dataset, compared with the particle filter with the same number of particles and a fixed resampling strategy (MIT-2 curve).

To give a more detailed impression about the accuracy of our new mapping technique, Fig. 9 depicts maps learned from well-known and freely available [25] real robot datasets recorded at the University of Texas, at the University of Washington, at Belgioioso, and at the University of Freiburg. Each map was built using 30 particles to represent the posterior about maps and poses.

D. The Influence of the Odometry on the Proposal

This experiment is designed to show the advantage of the proposal distribution, which takes into account the odometry information to draw particles. In most cases, the purely laser-based proposal, like the one presented in our previous approach [11] is well-suited to predict the motion of the particles. However, in a few situations, the knowledge about the odometry information can be important to focus the proposal distribution. This is the case if only very poor features are available in the laser data that was used to compute the parameters of the Gaussian proposal approximation. For example, an open free space without any obstacles or a long featureless corridor can lead to high variances in the computed proposal that is only based on laser range-finder data. Fig. 10 illustrates this effect based on simulated laser data.

In a further experiment, we simulated a short-range laser scanner (like, e.g., the Hokuyo URG scanner). Due to the maximum range of 4 m, the robot was unable to see the end of the corridor in most cases. This results in a high pose uncertainty in the direction of the corridor. We recorded several trajectories in this environment, and used them to learn maps with and without considering the odometry when computing the proposal distribution. In this experiment, the approach considering the odometry succeeded in 100% of cases to learn a topologically

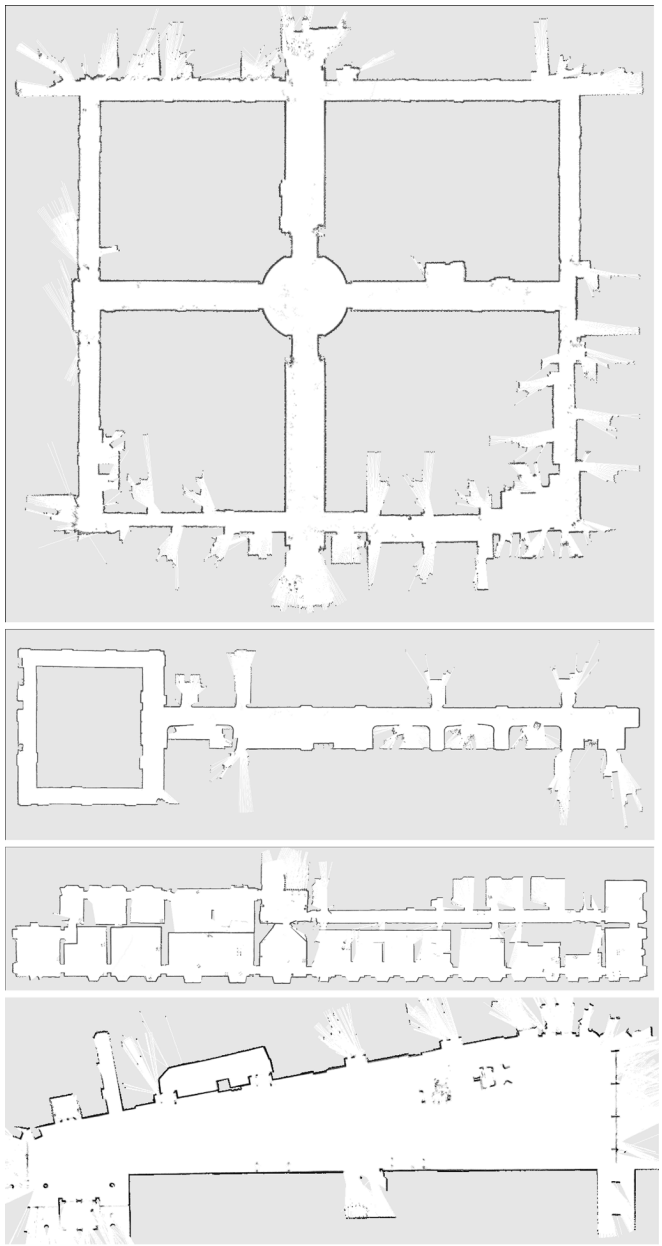

Fig. 9. Maps from the ACES building at University of Texas, the 4th floor of Sieg Hall at the University of Washington, the Belgioioso building, and Building 101 at the University of Freiburg.

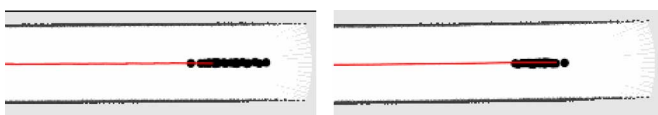

Fig. 10. Effect of considering the odometry in the computation of the proposal on the particle cloud. The left image depicts the particle distribution if only the laser range-finder data is used. By taking into account the odometry when computing the proposal distribution, the particles can be drawn in a more accurate manner. As can be seen in the right image, the particle cloud is more focused, because it additionally incorporates the odometry information.

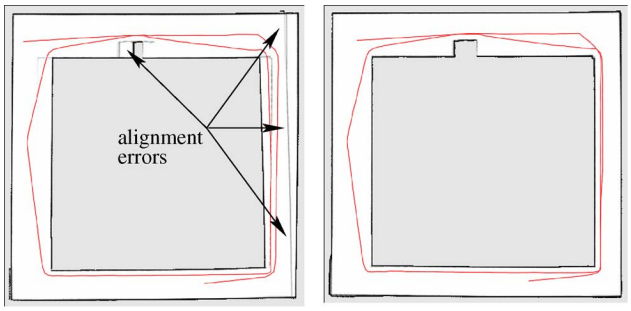

Fig. 11. Different mapping results for the same data set obtained using the proposal distribution which ignores the odometry (left image), and which considers the odometry when drawing the next generation of particles (right image).

correct map. Contrast to that our previous approach, which does not take into account the odometry, and succeeded only in 50% of cases. This experiment indicates the importance of the improved proposal distribution. Fig. 11 depicts typical maps obtained with the different proposal distributions during this experiment. The left map contains alignment errors caused by the high pose uncertainty in the direction of the corridor. In contrast to that, a robot that also takes into account the odometry was able to maintain the correct pose hypotheses. A typical example is depicted in the right image.

Note that by increasing the number of particles, both approaches are able to map the environment correctly in 100% of cases, but since each particle carries its own map, it is of utmost importance to keep the number of particles as low as possible. Therefore, this improved proposal is a means to limit the number of particles during mapping with RBPFs.

E. Situations in Which the Scan-Matcher Fails

As reported in Section III, it can happen that the scan-matcher is unable to find a good pose estimate based on the laser range data. In this case, we sample from the raw odometry model to obtain the next generation of particles. In most tested indoor datasets, however, such a situation never occurred at all. In the MIT dataset, this effect was observed once, due to a person walking directly in front of the robot while the robot was moving though a corridor that mainly consists of glass panes.

In outdoor datasets, such a situation can occur if the robot moves through large open spaces, because in this case, the laser range finder mainly reports maximum-range readings. While mapping the Freiburg campus, the scan-matcher also reported such an error at one point. In this particular situation, the robot entered the parking area (in the upper part of the map, compare Fig. 5). On that day, all cars were removed from the parking area due to construction work. As a result, no cars or other objects caused reflections of the laser beams, and most parts of the scan consisted of maximum-range readings. In such a situation, the odometry information provides the best pose estimate, and this information is used by our mapping system to predict the motion of the vehicle.

F. Runtime Analysis

In this last experiment, we analyze the memory and computational resources needed by our mapping system. We used a standard PC with a 2.8-GHz processor. We recorded the average memory usage and execution time, using the default parameters that allows our algorithm to learn correct maps for nearly all real-world datasets provided to us. In this setting, 30 particles are used to represent the posterior about maps and poses, and a new observation, consisting of a full laser range scan, is integrated whenever the robot moved more than 0.5 m or rotated more than 25°. The Intel Research Lab dataset (see Fig. 4) contains odometry and laser range readings which have been recorded over 45 min. Our implementation required 150 MB of memory to store all the data, using maps with a size of approximately 40 m by 40 m and a grid resolution of 5 cm. The overall time to correct the log file using our software was less than 30 min. This means that the time to record a log file is around 1.5 times longer than the time to correct the log file. Table III depicts the average execution time for the individual operations.

TABLE III
AVERAGE EXECUTION TIME USING A STANDARD PC

Operation	Average Execution Time
Computation of the proposal distribution, the weights, and the map update	1910 ms
Test if resampling is required	41 ms
Resampling	244 ms

VII. RELATED WORK

Mapping techniques for mobile robots can be roughly classified according to the map representation and the underlying estimation technique. One popular map representation is the occupancy grid [12]. Whereas such grid-based approaches are computationally expensive and typically require a huge amount of memory, they are able to represent arbitrary objects. Feature-based representations are attractive because of their compactness. However, they rely on predefined feature extractors, which assumes that some structures in the environments are known in advance.

The estimation algorithms can be roughly classified according to their underlying basic principle. The most popular approaches are EKFs, maximum-likelihood (ML) techniques, sparse extended information filters (SEIFs), smoothing techniques, and RBPFs. The effectiveness of the EKF approaches comes from the fact that they estimate a fully correlated posterior over landmark maps and robot poses [26], [27]. Their weakness lies in the strong assumptions that have to be made on both the robot motion model and the sensor noise. Moreover, the landmarks are assumed to be uniquely identifiable. There exist techniques [28] to deal with unknown data association in the SLAM context; however, if these assumptions are violated, the filter is likely to diverge [29]. Similar observations have been reported by Julier *et al.* [30], as well as by Uhlmann [31]. The unscented Kalman filter described in [30] is one way of better dealing with the nonlinearities in the motion model of the vehicle.

A popular ML algorithm computes the most likely map, given the history of sensor readings, by constructing a network of relations that represents the spatial constraints between the poses of the robot [4], [19], [32], [33]. Gutmann *et al.* [4] proposed

an effective way for constructing such a network and for detecting loop closures, while running an incremental ML algorithm. When a loop closure is detected, a global optimization on the network of relation is performed. Recently, Hähnel *et al.* [34], proposed an approach which is able to track several map hypotheses using an association tree. However, the necessary expansions of this tree can prevent the approach from being feasible for real-time operation.

Thrun *et al.* [35] proposed a method to correct the poses of robots based on the inverse of the covariance matrix. The advantage of the SEIFs is that they make use of the approximative sparsity of the information matrix, and in this way, can perform predictions and updates in constant time. Eustice *et al.* [36] presented a technique to make use of exactly sparse information matrices in a delayed-state framework. Paskin [37] presented a solution to the SLAM problem using thin junction trees. In this way, he is able to reduce the complexity compared with the EKF approaches, since thinned junction trees provide a linear-time filtering operation.

Folkessen *et al.* [38] proposed an effective approach for dealing with symmetries and invariants that can be found in landmark-based representation. This is achieved by representing each feature in a low-dimensional space (measurement subspace) and in the metric space. The measurement subspace captures an invariant of the landmark, while the metric space represents the dense information about the feature. A mapping between the measurement subspace and the metric space is dynamically evaluated and refined as new observations are acquired. Such a mapping can take into account spatial constraints between different features. This allows the authors to consider these relations for updating the map estimate.

Very recently, Dellaert proposed a smoothing method called square-root smoothing and mapping [39]. It has several advantages compared with EKF, since it better covers the nonlinearities and is faster to compute. In contrast to SEIFs, it furthermore provides an exactly sparse factorization of the information matrix.

In work by Murphy, Doucet, and colleagues [2], [8], RBPF have been introduced as an effective means to solve the SLAM problem. Each particle in an RBPF represents a possible robot trajectory and a map. The framework has been subsequently extended by Montemerlo *et al.* [7], [40] for approaching the SLAM problem with landmark maps. To learn accurate grid maps, RBPFs have been used by Eliazar and Parr [3] and Hähnel *et al.* [5]. Whereas the first work describes an efficient map representation, the second presents an improved motion model that reduces the number of required particles. Based on the approach of Hähnel *et al.*, Howard presented an approach to learn grid maps with multiple robots [41]. The focus of this work lies in how to merge the information obtained by the individual robots, and not in how to compute better proposal distributions.

Bosse *et al.* [42] describe a generic framework for SLAM in large-scale environments. They use a graph structure of local maps with relative coordinate frames, and always represent the uncertainty with respect to a local frame. In this way, they are able to reduce the complexity of the overall problem. In this context, Modayil *et al.* [43] presented a technique which combines metrical SLAM with topological SLAM. The topology is used to solve the loop-closing problem, whereas metric information is used to build up local structures. Similar ideas have been realized by Lisien *et al.* [44], which introduce a hierarchical map in the context of SLAM.

The work described in this paper is an improvement of the algorithm proposed by Hähnel *et al.* [5]. Instead of using a fixed proposal distribution, our algorithm computes an improved proposal distribution on a per-particle basis on the fly. This allows us to directly use the information obtained from the sensors while evolving the particles. The work presented here is also an extension of our previous approach [11], which lacks the ability to incorporate the odometry information into the proposal. Especially, in critical situations in which only poor laser features for localization are available, our current approach performs better than our previous one.

The computation of the proposal distribution is done in a similar way as in FastSLAM-2, presented by Montemerlo *et al.* [6]. In contrast to FastSLAM-2, our approach does not rely on predefined landmarks, and uses raw laser range-finder data to acquire accurate grid maps. Particle filters using proposal distributions that take into account the most recent observation are also called look-ahead particle filters. Moralez-Menéndez *et al.* [14] proposed such a method to more reliably estimate the state of a dynamic system where accurate sensors are available.

The advantage of our approach is twofold. First, our algorithm draws the particles in a more effective way. Second, the highly accurate proposal distribution allows us to use the effective sample size as a robust indicator to decide whether or not a resampling has to be carried out. This further reduces the risk of particle depletion.

VIII. Conclusions

In this paper, we presented an improved approach to learning grid maps with RBPFs. Our approach computes a highly accurate proposal distribution based on the observation likelihood of the most recent sensor information, the odometry, and a scan-matching process. This allows us to draw particles in a more accurate manner, which seriously reduces the number of required samples. Additionally, we apply a selective resampling strategy based on the effective sample size. This approach reduces the number of unnecessary resampling actions in the particle filter, and thus substantially reduces the risk of particle depletion.

Our approach has been implemented and evaluated on data acquired with different mobile robots equipped with laser range scanners. Tests performed with our algorithm in different large-scale environments have demonstrated its robustness and ability to generate high-quality maps. In these experiments, the number of particles needed by our approach often was one order of magnitude smaller, compared with previous approaches.

Acknowledgment

The authors would like to acknowledge M. Bosse and J. Leonard for providing them the dataset of the MIT Killian Court, P. Beeson for the ACES dataset, and D. Hähnel for the Intel Research Lab, the Belgioioso, and the Sieg Hall datasets.

REFERENCES

[1] G. Dissanayake, H. Durrant-Whyte, and T. Bailey, "A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem," in *Proc. IEEE Int. Conf. Robot. Autom.*, San Francisco, CA, 2000, pp. 1009–1014.

[2] A. Doucet, J. de Freitas, K. Murphy, and S. Russel, "Rao-Blackwellized particle filtering for dynamic Bayesian networks," in *Proc. Conf. Uncertainty Artif. Intell.*, Stanford, CA, 2000, pp. 176–183.

[3] A. Eliazar and R. Parr, "DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks," in *Proc. Int. Conf. Artif. Intell.*, Acapulco, Mexico, 2003, pp. 1135–1142.

[4] J.-S. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments," in *Proc. IEEE Int. Symp. Comput. Intell. Robot. Autom.*, Monterey, CA, 1999, pp. 318–325.

[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. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Las Vegas, NV, 2003, pp. 206–211.

[6] M. Montemerlo, S. T. D. Koller, and B. Wegbreit, "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *Proc. Int. Conf. Artif. Intell.*, Acapulco, Mexico, 2003, pp. 1151–1156.

[7] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to simultaneous localization and mapping," in *Proc. Nat. Conf. Artif. Intell.*, Edmonton, AB, Canada, 2002, pp. 593–598.

[8] K. Murphy, "Bayesian map learning in dynamic environments," in *Proc. Conf. Neural Inf. Process. Syst.*, Denver, CO, 1999, pp. 1015–1021.

[9] S. Thrun, "An online mapping algorithm for teams of mobile robots," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 335–363, 2001.

[10] R. van der Merwe, N. de Freitas, A. Doucet, and E. Wan, "The unscented particle filter," Cambridge Univ. Eng. Dept., Cambridge, U.K., Tech. Rep. CUED/F-INFENG/TR380, 2000.

[11] G. Grisetti, C. Stachniss, and W. Burgard, "Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, 2005, pp. 2443–2448.

[12] H. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI Mag.*, pp. 61–74, 1988.

[13] A. Doucet, N. de Freitas, and N. Gordan, Eds., *Sequential Monte-Carlo Methods in Practice*. New York: Springer-Verlag, 2001.

[14] R. Morales-Menéndez, N. de Freitas, and D. Poole, "Real-time monitoring of complex industrial processes with particle filters," in *Proc. Conf. Neural Inf. Process. Syst.*, Vancover, BC, Canada, 2002, pp. 1433–1440.

[15] M. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," Imperial College, Dept. Math, London, U.K., Tech. Rep., 1997.

[16] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo localization for mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Leuven, Belgium, 1998, pp. 99–141.

[17] A. Doucet, "On sequential simulation-based methods for Bayesian filtering," Univ. Cambridge, Dept. Eng., Signal Process. Group, Cambridge, U.K., Tech. Rep., 1998.

[18] J. Liu, "Metropolized independent sampling with comparisons to rejection sampling and importance sampling," *Statist. Comput.*, vol. 6, pp. 113–119, 1996.

[19] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *J. Auton. Robots*, vol. 4, pp. 333–349, 1997.

[20] M. Montemerlo, N. Roy, S. Thrun, D. Hähnel, C. Stachniss, and J. Glover, CARMEN—The Carnegie Mellon Robot Navigation Toolkit 2002 [Online]. Available: http://carmen.sourceforge.net

[21] N. Roy, M. Montemerlo, and S. Thrun, "Perspectives on standardization in mobile robot programming," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Las Vegas, NV, 2003, pp. 2436–2441.

[22] S. Thrun, W. Burgard, and D. Fox, "Robot perception," in *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005, pp. 171–172.

[23] ——, "Robot motion," in *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005, pp. 121–123.

[24] C. Stachniss and G. Grisetti, "Mapping results obtained with Rao-Blackwellized particle filters," 2004 [Online]. Available: http://www.informatik.uni-freiburg.de/~stachnis/research/rbpfmapper/

[25] A. Howard and N. Roy, "The Robotics Data Set Repository (Radish)," 2003 [Online]. Available: http://radish.sourceforge.net/

[26] J. Leonard and H. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Trans. Robot. Autom.*, vol. 7, no. 3, pp. 376–382, Jun. 1991.

[27] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*, I. Cox and G. Wilfong, Eds. New York: Springer-Verlag, 1990, pp. 167–193.

[28] J. Neira and J. Tardós, "Data association in stochastic mapping using the joint compatibility test," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 890–897, Dec. 2001.

[29] U. Frese and G. Hirzinger, "Simultaneous localization and mapping—A discussion," in *Proc. IJCAI Workshop Reasoning with Uncertainty Robot.*, Seattle, WA, 2001, pp. 17–26.

[30] S. Julier, J. Uhlmann, and H. Durrant-Whyte, "A new approach for filtering nonlinear systems," in *Proc. Amer. Control Conf.*, Seattle, WA, 1995, pp. 1628–1632.

[31] J. Uhlmann, "Dynamic map building and localization: New theoretical foundations," Ph.D. dissertation, Univ. Oxford, Oxford, U.K., 1995.

[32] T. Duckett, S. Marsland, and J. Shapiro, "Fast, on-line learning of globally consistent maps," *J. Auton. Robots*, vol. 12, no. 3, pp. 287–300, 2002.

[33] U. Frese, P. Larsson, and T. Duckett, "A multilevel relaxation algorithm for simultaneous localization and mapping," *IEEE Trans. Robot.*, vol. 21, no. 1, pp. 1–12, Feb. 2005.

[34] D. Hähnel, W. Burgard, B. Wegbreit, and S. Thrun, "Towards lazy data association in SLAM," in *Proc. Int. Symp. Robot. Res.*, Siena, Italy, 2003, pp. 421–431.

[35] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," *Int. J. Robot. Res.*, vol. 23, no. 7/8, pp. 693, 716, 2004.

[36] R. Eustice, H. Singh, and J. Leonard, "Exactly sparse delayed-state filters," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, 2005, pp. 2428–2435.

[37] M. Paskin, "Thin junction tree filters for simultaneous localization and mapping," in *Proc. Int. Conf. Artif. Intell.*, Acapulco, Mexico, 2003, pp. 1157–1164.

[38] J. Folkesson, P. Jensfelt, and H. Christensen, "Vision SLAM in the measurement subspace," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2005, pp. 325–330.

[39] F. Dellaert, "Square root SAM," in *Proc. Robot: Sci. Syst.*, Cambridge, MA, 2005, pp. 177–184.

[40] M. Montemerlo and S. Thrun, "Simultaneous localization and mapping with unknown data association using FastSLAM," in *Proc. IEEE Int. Conf. Robot. Autom.*, Taipei, Taiwan, R.O.C., 2003, pp. 1985–1991.

[41] A. Howard, "Multi-robot simultaneous localization and mapping using particle filters," in *Proc. Robot.: Sci. Syst.*, Cambridge, MA, 2005, pp. 201–208.

[42] M. Bosse, P. Newman, J. Leonard, and S. Teller, "An ALTAS framework for scalable mapping," in *Proc. IEEE Int. Conf. Robot. Autom.*, Taipei, Taiwan, R.O.C., 2003, pp. 1899–1906.

[43] J. Modayil, P. Beeson, and B. Kuipers, "Using the topological skeleton for scalable global metrical map-building," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sendai, Japan, 2004, pp. 1530–1536.

[44] B. Lisien, D. S. D. Morales, G. Kantor, I. Rekleitis, and H. Choset, "Hierarchical simultaneous localization and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Las Vegas, NV, 2003, pp. 448–453.

Giorgio Grisetti received the M.Sc. degree in computer engineering from the University of Rome, Rome, Italy, in 2001, and the Ph.D. degree in 2006 from the University of Rome "La Sapienza," Rome, Italy.

He is currently a Postdoctoral Researcher with the research lab for Autonomous Intelligent Systems, University of Freiburg, Freiburg, Germany. His research interests lie in the areas of SLAM, mobile robot localization, and probabilistic state estimation. His Ph.D. dissertation focused on SLAM using Rao-Blackwellized particle filters.

Cyrill Stachniss received the M.Sc. degree in computer science in 2002, and the Ph.D. degree in 2006, both from the University of Freiburg, Freiburg, Germany. From 1997 to 2000, he studied computer science (B.Sc.) and physics (B.Sc.) at the University of Marburg, Marburg, Germany.

He is currently with Eidgenössische Technische Hochschule (ETH), Zurich, Switzerland, as a Postdoctoral Researcher. His research interests lie in the areas of mobile robot exploration, SLAM, and collision avoidance. His Ph.D. dissertation focused on exploration and mapping with mobile robots.

Wolfram Burgard (M'97) studied computer science at the University of Dortmund, Dortmund, Germany, and received his Ph.D. degree from the Department of Computer Science, University of Bonn, Bonn, Germany, in 1991.

Since 1999, has been an Associate Professor with the Department of Computer Science, University of Freiburg, Freiburg, Germany, where he heads the research laboratory for Autonomous Intelligent Systems. His areas of interest lie in artificial intelligence and robotics. They cover mobile robot navigation, multirobot systems, state estimation, human–robot interaction, activity monitoring, and networked robots.