Particle Filters for Robot Navigation

Cyrill Stachniss
University of Freiburg
stachnis@informatik.uni-freiburg.de

Wolfram Burgard
University of Freiburg
burgard@informatik.uni-freiburg.de

the essence of knowledge
Boston — Delft

Foundations and Trends® in Robotics

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

C. Stachniss and W. Burgard. *Particle Filters for Robot Navigation*. Foundations and Trends® in Robotics, vol. 3, no. 4, pp. 211–282, 2012.

This Foundations and Trends® issue was typeset in LaTeX using a class file designed by Neal Parikh. Printed on acid-free paper.

ISBN: 978-1-60198-759-4

Editorial Scope

Topics

Foundations and Trends® in Robotics publishes survey and tutorial articles in the following topics:

- Mathematical modelling
- Kinematics
- Dynamics
- Estimation methods
- Artificial intelligence in robotics
- Software systems and architectures
- Sensors and estimation
- Planning and control
- Human-robot interaction
- Industrial robotics
- Service robotics

Information for Librarians

Foundations and Trends® in Robotics, 2012, Volume 3, 4 issues. ISSN paper version 1935-8253. ISSN online version 1935-8261. Also available as a combined paper and online subscription.

Foundations and Trends® in Robotics
Vol. 3, No. 4 (2012) 211–282

DOI: 10.1561/2300000013

Particle Filters for Robot Navigation

Cyrill Stachniss
University of Freiburg
stachnis@informatik.uni-freiburg.de

Wolfram Burgard
University of Freiburg
burgard@informatik.uni-freiburg.de

Contents

Abstract

Autonomous navigation is an essential capability for mobile robots. In order to operate robustly, a robot needs to know what the environment looks like, where it is in its environment, and how to navigate in it. This work summarizes approaches that address these three problems and that use particle filters as their main underlying model for representing beliefs. We illustrate that these filters are powerful tools that can robustly estimate the state of the robot and its environment and that it is also well-suited to make decisions about how to navigate in order to minimize the uncertainty of the joint belief about the robot's position and the state of the environment.

C. Stachniss and W. Burgard. *Particle Filters for Robot Navigation*. Foundations and Trends® in Robotics, vol. 3, no. 4, pp. 211–282, 2012.

DOI: 10.1561/2300000013.

1

Particle Filters for Robot Navigation

The ability to reliably navigate is an essential capability for autonomous robots. In order to perform effective navigation tasks, robots typically need to know what the environment looks like, where they are in the environment, and how to reach a target location. Thus, models of the environment play an important role for effective navigation. Learning maps has therefore been a major research focus in the robotics community over the last decades.

Three main capabilities are needed for traveling through an environment and for learning an appropriate representation. These are *mapping*, *localization*, and *motion generation*. Mapping is the problem of integrating the information gathered with the robot's sensors into a given representation. It can be described by the question "What does the world look like?" In contrast to this, localization is the problem of estimating the pose, *i.e.*, the position and heading, of the robot relative to a map. In other words, the robot has to answer the question, "Where am I?" Finally, the motion generation problem involves the question of where to go and how to efficiently calculate a path to guide a vehicle to that location. Expressed as a simple question, this problem can be described as, "Where should I go and how to reach that location?"

Unfortunately, these three tasks cannot be solved independently of each other. Before a robot can answer the question of what the environment looks like given a set of observations, it needs to know from which locations these observations have been made. At the same time, it is hard to estimate the current position of a vehicle without a map. Planning a path to a goal location is also tightly coupled with the knowledge of what the environment looks like as well as with the information about the current pose of the robot.

It is important to acknowledge that robots operate in unpredictable environments and that the sensors and the actuation of robots are inherently uncertain. Therefore, robots need the ability to deal with uncertainty and to explicitly model it, even for performing basic tasks. Particle filters are one way for performing state estimation in the presence of uncertainty. They offer a series of attractive capabilities, including the ability to deal with non-Gaussian distributions and nonlinear sensor and motion models. This article describes particle filter-based systems developed by the authors in the context of robot navigation.

1.1 The Bayes Filter

Before introducing the particle filter, we start with the Bayes filter as the particle filter is a special implementation of the Bayes filter. The Bayes filter is a general algorithm for estimating a belief given control commands and observations. The goal is to estimate the distribution about the current state x_t at time t given all commands $u_{1:t} = u_1, \ldots, u_t$ and observations $z_{1:t} = z_1, \ldots, z_t$. The Bayes filter performs this estimation in a recursive manner using a prediction step that takes into account the current control command and a correction step that uses the current observation.

In the prediction step, the filter computes a predicted belief $\overline{bel}(x_t)$ at time t based on the previous belief $bel(x_{t-1})$ and a model that describes how the command u_t changes the state from $t-1$ to t. This model $p(x_t \mid x_{t-1}, u_t)$ is called transition model or motion model.

In the correction step, the filter corrects the predicted belief by taking into account the observation. It does so by multiplying the predicted belief $\overline{bel}(x_t)$ with the observation model $p(z_t \mid x_t)$ and a normalizing constant η. The normalizer ensures that the integral over all possible states x_t equals to 1 so that we obtain a probability distribution.

Algorithm 1.1 depicts the Bayes filter algorithm with the prediction step in Line 2 and the correction step in Line 3. The algorithm computes the belief at time t based on the previous belief at $t-1$. Thus, an initial belief at time $t = 0$, the so called prior belief, serves as a starting point for the estimation process. If no prior knowledge is available, the $bel(x_0)$ is a uniform distribution.

The Bayes filter can be derived formally by using only Bayes' rule, Markov assumptions, and the law of total probability:

$$
\begin{aligned}
& bel(x_t) \\
& \overset{\text{Definition}}{=} \quad p(x_t \mid z_{1:t}, u_{1:t}) && (1.1) \\
& \overset{\text{Bayes' rule}}{=} \quad \eta\, p(z_t \mid x_t, z_{1:t-1}, u_{1:t})\, p(x_t \mid z_{1:t-1}, u_{1:t}) && (1.2) \\
& \overset{\text{Markov}}{=} \quad \eta\, p(z_t \mid x_t)\, p(x_t \mid z_{1:t-1}, u_{1:t}) && (1.3) \\
& \overset{\text{Total prob.}}{=} \quad \eta\, p(z_t \mid x_t) \int p(x_t \mid x_{t-1}, z_{1:t-1}, u_{1:t}) && (1.4) \\
& \qquad\qquad p(x_{t-1} \mid z_{1:t-1}, u_{1:t})\, dx_{t-1} && (1.5) \\
& \overset{\text{Markov}}{=} \quad \eta\, p(z_t \mid x_t) \\
& \qquad\qquad \int p(x_t \mid x_{t-1}, u_t)\, p(x_{t-1} \mid z_{1:t-1}, u_{1:t})\, dx_{t-1} && (1.6) \\
& \overset{\text{Ignoring } u_t}{=} \quad \eta\, p(z_t \mid x_t) \\
& \qquad\qquad \int p(x_t \mid x_{t-1}, u_t)\, p(x_{t-1} \mid z_{1:t-1}, u_{1:t-1})\, dx_{t-1} && (1.7) \\
& \overset{\text{Definition}}{=} \quad \underbrace{\eta\, p(z_t \mid x_t) \underbrace{\int p(x_t \mid x_{t-1}, u_t)\, bel(x_{t-1})\, dx_{t-1}}_{\text{prediction step}}}_{\text{correction step}} && (1.8)
\end{aligned}
$$

This derivation shows that the belief $bel(x_t)$ can be estimated recursively based on the previous belief $bel(x_{t-1})$ and that is given by the product of the prediction step and the correction step.

The Bayes filter makes several assumptions in order to derive the recursive update scheme. It makes the assumption that given we know the state x_t, the observation z_t is independent from the previous observations and controls, see (1.3). In the same way, it assumes that the state x_t is independent from all observations and controls collected up to $t-1$ if we know x_{t-1}, see (1.3). Finally, it assumes that estimating the state of the system at time $t-1$ is independent from the future control command u_t, see (1.7).

Algorithm 1.1 The Bayes filter algorithm

Input: u_t, z_t, $bel(x_{t-1})$
1: **for all** x_t **do**
2: $\overline{bel}(x_t) = \int p(x_t \mid x_{t-1}, u_t)\, bel(x_{t-1})\, dx_{t-1}$
3: $bel(x_t) = \eta\, p(z_t \mid x_t)\, \overline{bel}(x_t)$
4: **end**
5: **return** $bel(x_t)$

1.2 The Particle Filter

The root of particle filters can be traced back for around 60 years [32] but they have become popular only in the last two decades. Particle filters represent a posterior through a set of samples or particles. Each sample is best thought as a concrete guess of what the true value of the state may be. By maintaining a set of samples, *i.e.*, a set of different state hypotheses, the sample set approximates the posterior distribution.

The particle filter is an implementation of the Bayes filter. As the Bayes filter, it allows for maintaining a probability distribution that is updated based on the commands that are executed by the robot and based on the observations that the robot acquires. The particle filter is a nonparametric Bayes filter as it presents the belief not in closed form but using a finite number of parameters. It models the belief by samples, which represent possible states the system might be in. For example, if we aim at estimating the pose of a robot, the particles model all possible positions and orientations the robot may be located at given our current knowledge.

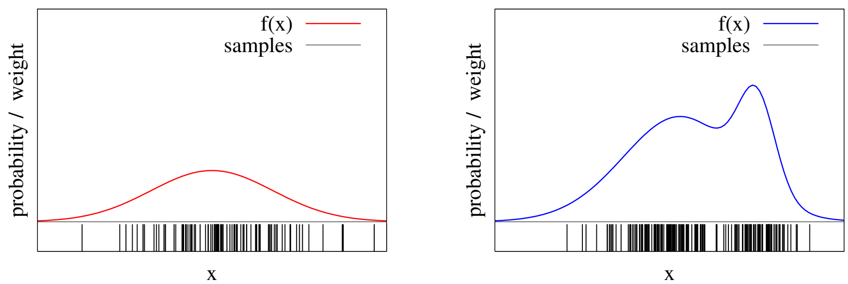

Figure 1.1: Two functions and their approximations by samples with uniform weights. The samples are illustrated by the vertical bars below the two functions.

The belief at time t is represented by a set S_t of N weighted random samples

$$S_t \quad = \quad \left\{ \left\langle x_t^{[i]}, w_t^{[i]} \right\rangle \,\middle|\, i = 1, \ldots, N \right\}, \tag{1.9}$$

where $x_t^{[i]}$ is the state vector of the i-th sample and $w_t^{[i]}$ the corresponding weight. The weight is a non-zero value and the sum over all weights is 1. The sample set represents the distribution

$$p(x_t) \quad = \quad \sum_{i=1}^{N} w_t^{[i]} \delta_{x_t^{[i]}}(x_t), \tag{1.10}$$

where $\delta_{x_t^{[i]}}$ is the Dirac function in $x_t^{[i]}$. As a result of (1.10), the higher the sum of weights of samples that fall in one region of the space, the higher the likelihood that the true state lies in this region.

One interesting property of sample-based representations is the ability to approximate arbitrary distributions. This is an advantage over frequently used parametric models. For example, the ability to model multi-modal distributions by the set of samples is an advantage compared to Gaussian distributions. To illustrate such an approximation, Figure 1.1 depicts two distributions and their corresponding sample sets. In general, the more samples are used, the better the approximation is.

1.2.1 An Intuitive Explanation of the Particle Filter Algorithm

Whenever we are interested in estimating the state of a dynamic system over time, we can apply the particle filter algorithm for updating and maintaining a sample set given controls and observations. The algorithm allows us to recursively estimate the particle set S_t based on the estimate S_{t-1} of the previous time step. The particle filter can be summarized by the following three steps:

1. **Sampling:** Create the next generation $\bar{S}_t$ of particles based on the previous set S_{t-1} of samples. In this step, we draw samples from a so-called proposal distribution. The proposal distribution thus describes how the state evolves.

   If we choose the motion model $p(x_t \mid x_{t-1}, u_t)$ starting with S_{t-1} as our proposal distribution, this sampling process corresponds to the prediction step of the Bayes filter. This becomes clear if we consider that each sample in S_{t-1} corresponds to a possible state hypothesis at time $t-1$. Drawing for every state hypothesis $x_{t-1}^{[i]}$ a new state x' according to $p(x' \mid x_{t-1}^{[i]}, u_t)$, generates the predicted belief $\overline{bel}(x_t)$.

2. **Importance Weighting:** Compute the importance weight $w_t^{[i]}$ for each sample in $\bar{S}_t$.

   Continuing the analogy to the Bayes filter, this operation corresponds to the correction step. By assigning to each state hypothesis of the predicted belief the weight $w_t^{[i]} = \eta\, p(z_t \mid x_t^{[i]})$, we obtain $bel(x_t)$ by (1.10).

3. **Resampling:** Draw N samples from the current sample set with replacement. Thereby, the likelihood to draw a particle is proportional to its weight. The new set S_t is given by the drawn particles and their weights are set to $1/N$.

   The resampling operation has no analogous step in the Bayes filter algorithm and thus can be confusing at first sight. The resampling step, however, is an important element of all particle filter implementations. The resampling step creates a new sample set that has the same size as the previous one. Before the resampling step, the particles are distributed according to the predicted belief $\overline{bel}(x_t)$ whereas they are distributed according to $bel(x_t)$ after resampling. This operation tends to eliminate samples with a low likelihood after the correction step and thus reorganizes the sample set according to the posterior $bel(x_t)$.

1.2.2 A Formal Explanation of the Particle Filter Algorithm

In addition to the intuitive explanation of the particle filter, we can also introduce the algorithm more formally. The goal is to obtain a sampled representation of our belief, *i.e.*, the target probability distribution. In each step, we can draw samples in order to obtain the generation of particles representing the distribution that was used for sampling. In general, the target probability distribution $p(x)$ for sampling particles is not known or not in a suitable form for sampling. It is, however, possible to draw the samples from a distribution $\pi(x)$ that is different from the distribution $p(x)$ that we want to approximate. A technique to do that in a sound way is *importance sampling*. The key idea of importance sampling is to draw the samples from π but use a weight associated to each sample that considers the difference between π and p.

In importance sampling, we are faced with the problem of computing the expectation that x, which follows the probability density function p, lies within a region A. Let B be an indicator function, which returns 1 if its argument is true and 0 otherwise. We can express the expectation that $x \in A$ by

$$\begin{aligned} E_p[B(x \in A)] &= \int p(x) B(x \in A)\, dx && (1.11) \\ &= \int \frac{p(x)}{\pi(x)} \pi(x) B(x \in A)\, dx, && (1.12) \\ &= E_\pi[w(x) B(x \in A)] && (1.13) \end{aligned}$$

with $w(x) := p(x)/\pi(x)$. The factor $w(x)$ can be seen as a weighting factor that accounts for the difference between the probability density functions p and π at x. This means that even though we aim at

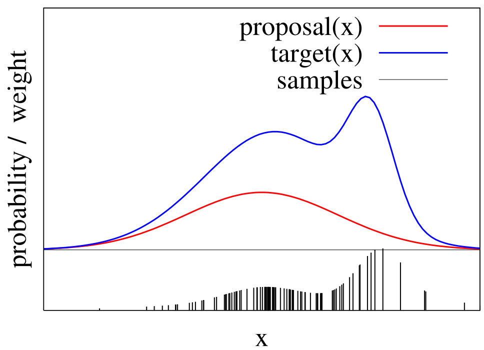

Figure 1.2: The goal is to approximate the target distribution by samples. The samples are drawn from the proposal distribution and weighted according to (1.14). After weighting, the resulting sample set is an approximation of the target distribution.

creating samples from p, we can draw the samples from a different density function π and weight each sample according to w. This holds as long as $p(x) > 0$ always implies that $\pi(x) > 0$. Otherwise, the state x could never be sampled. The function p is typically called the *target distribution* and π the *proposal distribution*. An example that depicts a weighted set of samples in case the proposal is different from the target distribution is shown in Figure 1.2. Note that the importance sampling principle requires that we can evaluate the target distribution in a point-wise fashion. Otherwise, the computation of the weights would be impossible.

Let p be the posterior to estimate and π the proposal distribution that is used in Step 1 of the particle filter for sampling. Then, the importance weighting performed in Step 2 accounts for the fact that one draws from the proposal π by setting the weight of each particle to

$$w_t^{[i]} \quad = \quad \eta \, \frac{p(x_t^{[i]})}{\pi(x_t^{[i]})}, \tag{1.14}$$

where η is a normalizer that ensures that the weights sum up to 1. Thus, by dividing the target probability distribution $p(x_t^{[i]})$ by the proposal distribution $\pi(x_t^{[i]})$, both evaluated in $x_t^{[i]}$, we re-weight the samples to consider the differences between p and π.

As the third and final step, the particle filter performs resampling, which refers to drawing N samples from the weighted sample set with replacement and resetting all weights in the new sample set to $1/N$. The likelihood to draw a sample is proportional to its weight computed in (1.14) and the drawn set is the result of the particle filter iteration. The resampling step is an important part of the particle filter as it distributes the samples according to $bel(x_t)$. The operation tends to eliminate samples with a low likelihood after the correction step. Therefore, it can be seen as a "survival of the fittest" step that avoids that samples deplete into unlikely regions of the state space. One popular way to implement resampling is low-variance resampling. The key idea of low-variance resampling is to avoid drawing the samples independently of each other. Only the first sample for the new set is drawn randomly and the other samples are drawn deterministically given the first draw but still with a probability proportional to the importance weight. This has two advantages. First, if all the samples have the same importance weight, the input sample set is equivalent to the output sample set, *i.e.*, no samples are lost in the resampling process. Second, the overall complexity of the algorithm is linear in the number of samples. The algorithm for low-variance resampling is shown in Algorithm 1.2 and the overall algorithm for particle filtering is given in Algorithm 1.3

To see the recursive nature of the particle filter mathematically, we consider the full posterior $bel(x_{0:t})$ about the sequence of states $x_0, \ldots, x_t$. We obtain the recursive formula by:

$$
\begin{aligned}
& p(x_{0:t} \mid z_{1:t}, u_{1:t}) \\
& \overset{\text{Bayes' rule}}{=} \quad \eta\, p(z_t \mid x_{0:t}, z_{1:t-1}, u_{1:t})\, p(x_{0:t} \mid z_{1:t-1}, u_{1:t}) && (1.15) \\
& \overset{\text{Markov}}{=} \quad \eta\, p(z_t \mid x_t)\, p(x_{0:t} \mid z_{1:t-1}, u_{1:t}) && (1.16) \\
& \overset{\text{Product rule}}{=} \quad \eta\, p(z_t \mid x_t)\, p(x_t \mid x_{0:t-1}, z_{1:t-1}, u_{1:t}) \\
& \qquad p(x_{0:t-1} \mid z_{1:t-1}, u_{1:t}) && (1.17) \\
& \overset{\text{Markov}}{=} \quad \eta\, p(z_t \mid x_t)\, p(x_t \mid x_{t-1}, u_t) \\
& \qquad p(x_{0:t-1} \mid z_{1:t-1}, u_{1:t-1}), && (1.18)
\end{aligned}
$$

Algorithm 1.2 The low-variance resampling algorithm

Input: Weighted sample set $\left\{\left\langle \hat{x}_t^{[i]}, \hat{w}_t^{[i]} \right\rangle \,\middle|\, i = 1, \ldots, N\right\}$.

$S_t = \emptyset$
$r = \text{rand\_uniform}\left((0; 1/N)\right)$
$c = \hat{w}_t^{[1]}$
$i = 1$
for $n = 1$ to N **do**
  $U = r + (n-1)/N$
  while $U > c$
    $i = i + 1$
    $c = c + \hat{w}_t^{[i]}$
  end
  $S_t = S_t \cup \left\{\left\langle \hat{x}_t^{[i]}, 1/N \right\rangle\right\}$
end
return S_t

Algorithm 1.3 The particle filter algorithm

Input: Sample set S_{t-1} representing the belief at $t-1$, control u_t, observation z_t.

$\bar{S}_t = \emptyset$
for i=1 to N **do**
  draw $\hat{x} \sim \pi(x_t \mid x_{t-1}^{[i]}, z_t, u_t)$
  $\hat{w} = \eta \left[p(\hat{x} \mid x_{t-1}^{[i]}, z_t, u_t)\right] \left[\pi(\hat{x} \mid x_{t-1}^{[i]}, z_t, u_t)\right]^{-1}$
  $\bar{S}_t = \bar{S}_t \cup \{< \hat{x}, \hat{w} >\}$
end
$S_t = \emptyset$
for j=1 to N **do**
  draw sample $\hat{x}_t^{[i]}$ from $\bar{S}_t$ with probability proportional to $\hat{w}_t^{[i]}$
  $S_t = S_t \cup \left\{\left\langle \hat{x}_t^{[i]}, 1/N \right\rangle\right\}$
end
return S_t

where η is the normalizer resulting from Bayes' rule. Under the Markov assumption, we can transform the proposal as

$$\begin{aligned}\pi(x_{0:t} \mid z_{1:t}, u_{1:t}) &= \pi(x_t \mid x_{t-1}, z_t, u_t) \\ &\quad \pi(x_{0:t-1} \mid z_{1:t-1}, u_{1:t-1}). \qquad (1.19)\end{aligned}$$

The computation of the weights needs to be done according to (1.14). For the not normalized weights, this leads to

$$\begin{aligned}w_t &= \frac{p(x_{0:t} \mid z_{1:t}, u_{1:t})}{\pi(x_{0:t} \mid z_{1:t}, u_{1:t})} \qquad (1.20)\\ &\overset{\text{Bayes' rule}}{=} \frac{\eta\, p(z_t \mid x_t)\, p(x_t \mid x_{t-1}, u_t)}{\pi(x_{0:t} \mid z_{1:t}, u_{1:t})} \\ &\qquad p(x_{0:t-1} \mid z_{1:t-1}, u_{1:t-1}) \qquad (1.21)\\ &= \frac{\eta\, p(z_t \mid x_t)\, p(x_t \mid x_{t-1}, u_t)}{\pi(x_t \mid x_{t-1}, z_t, u_t)} \\ &\qquad \underbrace{\frac{p(x_{0:t-1} \mid z_{1:t-1}, u_{1:t-1})}{\pi(x_{0:t-1} \mid z_{1:t-1}, u_{1:t-1})}}_{w_{t-1}} \qquad (1.22)\\ &= \frac{\eta\, p(z_t \mid x_t)\, p(x_t \mid x_{t-1}, u_t)}{\pi(x_t \mid x_{t-1}, z_t, u_t)}\, w_{t-1}. \qquad (1.23)\end{aligned}$$

As can be seen from this derivation, the weight at time t is computed as the weight at $t-1$ times a ratio that results from the importance sampling step at time t.

Note that the particle filter algorithm does not specify the proposal distribution. If we choose the motion model $p(x_t \mid x_{t-1}, u_t)$ as the proposal distribution for the current time step, *i.e.*, $\pi(x_t \mid x_{t-1}, z_t, u_t)$, we obtain the following importance weight for the i-th sample

$$\begin{aligned}w_t^{[i]} &= \frac{\eta\, p(z_t \mid x_t^{[i]})\, p(x_t \mid x_{t-1}^{[i]}, u_t)}{p(x_t \mid x_{t-1}^{[i]}, u_t)}\, w_{t-1}^{[i]} \qquad (1.24)\\ &= \eta\, p(z_t \mid x_t^{[i]})\, w_{t-1}^{[i]} \qquad (1.25)\\ &\propto p(z_t \mid x_t^{[i]})\, w_{t-1}^{[i]}. \qquad (1.26)\end{aligned}$$

We compute the sample set at time t based on the set at $t-1$ and as the resampling step resets the weights of the whole set to $1/N$, (1.26) is equivalent to

$$w_t^{[i]} \propto p(z_t \mid x_t^{[i]}). \quad (1.27)$$

This derivation shows that by choosing the motion model to draw the next generation of particles, we have to use the observation model $p(z_t \mid x_t)$ to compute the individual weights.

1.3 Summary

We introduced particle filters as a nonparametric implementation of the recursive Bayes filter. They use a set of weighted samples for modeling a belief and can represent arbitrary distributions. Each iteration of the particle filter algorithm consists of three steps that are sequentially executed. First, samples are drawn from a proposal distribution and this step corresponds to the prediction step in the Bayes filter framework. Second, an importance weight is computed for each sample that accounts for the fact that the target distribution is different from the proposal distribution. This step typically implements the correction step of the Bayes filter. Finally, the resulting sample set is obtained by drawing the weighted samples with replacement. The probability of drawing a sample is proportional to its weight.

References

[1] M. Adams and J. Mullane. Laser and radar based robotic perception. *Foundations and Trends in Robotics*, 1(3):135–252, 2010.

[2] H. Andreasson, A. Treptow, and T. Duckett. Localization for mobile robots using panoramic vision, local features and particle filter. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2005.

[3] A.M. Bauer, K. Klasing, G. Lidoris, Q. Mühlbauer, F. Rohrmüller, S. Sosnowski, T. Xu, K. Kühnlenz, D. Wollherr, and M. Buss. The autonomous city explorer: Towards natural human-robot interaction in urban environments. *International Journal of Social Robotics*, 1(2):127–140, 2009.

[4] M. Bennewitz, C. Stachniss, W. Burgard, and S. Behnke. Metric localization with scale-invariant visual features using a single perspective camera. In H.I. Christiensen, editor, *European Robotics Symposium 2006*, volume 22 of *STAR Springer tracts in advanced robotics*, pages 143–157. Springer-Verlag Berlin Heidelberg, Germany, 2006.

[5] F. Bourgoult, A.A. Makarenko, S.B. Williams, B. Grocholsky, and F. Durrant-Whyte. Information based adaptive robotic exploration. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Lausanne, Switzerland, 2002.

[6] M. Buehler, K. Iagnemma, and S. Singh, editors. *The DARPA 2005 Grand Challenge: The Great Robot Race*. Number 36 in STAR Springer tracts in advanced robotics. Springer Verlag, 2006.

[7] M. Buehler, K. Iagnemma, and S. Singh, editors. *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*. Number 56 in STAR Springer tracts in advanced robotics. Springer Verlag, 2009.

[8] W. Burgard, A. Derr, D. Fox, and A.B. Cremers. Integrating global position estimation and position tracking for mobile robots: The dynamic markov localization approach. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 1998.

[9] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 476–481, San Francisco, CA, USA, 2000.

[10] A. Censi. An ICP variant using a point-to-line metric. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, Pasadena, CA, May 2008.

[11] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 1999.

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

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

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

[15] T. Duckett, S. Marsland, and J. Shapiro. Fast, on-line learning of globally consistent maps. *Journal of Autonomous Robots*, 12(3):287 – 300, 2002.

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

[17] P. Elinas and J.J. Little. σMCL: Monte-Carlo localization for mobile robots with stereo vision. In *Proc. of Robotics: Science and Systems (RSS)*, 2005.

[18] D. Fox. Adapting the sample size in particle filters through kld-sampling. *Int. Journal of Robotics Research*, 2003.

[19] D. Fox, W. Burgard, and S. Thrun. Active markov localization for mobile robots. *Journal of Robotics & Autonomous Systems*, 25:195–207, 1998.

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

[21] H.-M. Gross, A. Köning, C. Schröter, and H.-J. Böhme. Omnivision-based probabilistic self-localization for a mobile shopping assistant continued. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2003.

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

[23] L.P. Kaelbling, A.R. Cassandra, and J.A. Kurien. Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Osaka, Japan, 1996.

[24] J. Ko, B. Stewart, D. Fox, K. Konolige, and B. Limketkai. A practical, decision-theoretic approach to multi-robot mapping and exploration. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3232–3238, Las Vegas, NV, USA, 2003.

[25] S. Koenig and C. Tovey. Improved analysis of greedy mapping. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA, 2003.

[26] B. Kuipers and Y.-T. Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Journal of Robotics & Autonomous Systems*, 8:47–63, 1991.

[27] R. Kümmerle, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard. A navigation system for robots operating in crowded urban environments. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, Karlsruhe, Germany, 2013.

[28] S. Lenser and M. Veloso. Sensor resetting localization for poorly modelled mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2000.

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

[30] J.S. Liu. Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statist. Comput.*, 6:113–119, 1996.

[31] A.A. Makarenko, S.B. Williams, F. Bourgoult, and F. Durrant-Whyte. An experiment in integrated exploration. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Lausanne, Switzerland, 2002.

[32] N. Metropolis and S. Ulam. The Monte Carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.

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

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

[35] R. Morales-Menéndez, N. de Freitas, and D. Poole. Real-time monitoring of complex industrial processes with particle filters. In *Proc. of the Conf. on Neural Information Processing Systems (NIPS)*, pages 1433–1440, Vancover, Canada, 2002.

[36] H.P. Moravec and A.E. Elfes. High resolution maps from wide angle sonar. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, pages 116–121, St. Louis, MO, USA, 1985.

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

[38] M.K. Pitt and N. Shephard. Filtering via simulation: auxilary particle filters. Technical report, Department of Mathematics, Imperial College, London, 1997.

[39] J.M. Porta, J.J. Verbeek, and B.J.A. Kröse. Active appearance-based robot localization using stereo vision. *Journal of Autonomous Robots*, 18:59–80, 2005.

[40] R. Rocha, J. Dias, and A. Carvalho. Exploring information theory for vision-based volumetric mapping. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2409–2414, Edmonton, Canada, 2005.

[41] T. Röfer, T. Laue, and D. Thomas. Particle-filter-based self-localization using landmarks and directed lines. In A. Bredenfeld, A. Jacoff, I. Noda, and Y. Takahashi, editors, *RoboCup 2005: Robot Soccer World Cup IX*, pages 608–615, 2006.

[42] J. Röwekaemper, C. Sprunk, G.D. Tipaldi, C. Stachniss, P. Pfaff, and W. Burgard. On the position accuracy of mobile robot localization based on particle filters combined with scan matching. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012.

[43] N. Roy, W. Burgard, D. Fox, and S. Thrun. Coastal navigation – robot motion with uncertainty. In *Proceedings of the AAAI Fall Symposium: Planning with POMDPs*, Stanford, CA, USA, 1998.

[44] R. Sim, G. Dudek, and N. Roy. Online control policy optimization for minimizing map uncertainty during exploration. In *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, New Orleans, LA, USA, 2004.

[45] C. Stachniss and W. Burgard. Exploring unknown environments with mobile robots using coverage maps. In *Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, pages 1127–1132, Acapulco, Mexico, 2003.

[46] C. Stachniss, G. Grisetti, W. Burgard, and N. Roy. Evaluation of gaussian proposal distributions for mapping with rao-blackwellized particle filters. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, San Diego, CA, USA, 2007.

[47] C. Stachniss, D. Hähnel, and W. Burgard. Exploration with active loop-closing for FastSLAM. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1505–1510, Sendai, Japan, 2004.

[48] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaertand D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J.Schulte, and D. Schulz. Probabilistic algorithms and the interactive museum tour-guide robot minerva. *Int. Journal of Robotics Research*, 19(11), 2000.

[49] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, 128, 2001.

[50] R. van der Merwe, N. de Freitas, A. Doucet, and E. Wan. The unscented particle filter. Technical Report CUED/F-INFENG/TR380, Cambridge University Engineering Department, August 2000.

[51] G. Weiß, C. Wetzler, and E. von Puttkamer. Keeping track of position and orientation of moving indoor systems by correlation of range-finder scans. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 595–601, Munich, Germany, 1994.

[52] P. Whaite and F. P. Ferrie. Autonomous exploration: Driven by uncertainty. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):193–205, 1997.

[53] B. Yamauchi. Frontier-based exploration using multiple robots. In *Proc. of the Second International Conference on Autonomous Agents*, pages 47–53, Minneapolis, MN, USA, 1998.