10

Mobile Robot Localization using Particle Filters and Sonar Sensors

Antoni Burguera, Yolanda González and Gabriel Oliver
Universitat de les Illes Balears
Spain

1. Introduction

1.1 The localization problem

Nowadays, nearly all mobile robotic tasks require some knowledge of the robot location in the environment. For example, those tasks involving the robot to reach a specific target require knowledge about the current robot pose in order to plan a path to the goal. Also, exploration tasks require some estimate of the robot pose in order to decide whether a specific region has been already visited by the robot or not. The problem of computing the robot pose is known as the *mobile robot localization* problem.

The mobile robot localization problem appears in many flavours. In some cases, only a *qualitative* pose estimate is needed. For example, for high level spatial reasoning, the robot may only need to know if a certain area, such as a room, has been previously visited or not. This kind of localization is commonly named *weak localization*. In some other cases, *quantitative* pose estimates with respect to a fixed reference frame are required. For example, to build metric maps, such as occupancy grids, the robot needs accurate numerical estimates of its pose in the space. This approach to localization is usually referred to as *strong localization*.

Both weak and strong localization problems can be defined in a global or in a local context, constituting the so called *global localization* and *local localization* problems respectively. The former refers to the obtention of the robot pose without an *a priori* estimate of its location. It is called global localization by analogy with global function minimization, whereby an optimum must be found without a reliable initial guess. On the contrary, local localization, sometimes named *pose maintenance*, refers to a continuous refinement of the robot pose, starting with an initial guess.

This chapter focuses on the strong localization problem in the local context. From now on, in the context of this document, the terms localization and mobile robot localization will refer to the strong localization problem in the local context. A common approach to confront this localization problem is the use of *exteroceptive* sensors, such as range finders or cameras, measuring the external environment. Exteroceptive sensor data is correlated at subsequent robot poses to compute displacement estimates, usually based on initial guesses provided by *proprioceptive* sensors, such as odometers or inertial units. As a consequence of this, the quality of the pose estimates is strongly related to the quality of the measurements provided by the exteroceptive sensors.

Source: Advances in Sonar Technology, Book edited by: Sergio Rui Silva,
ISBN 978-3-902613-48-6, pp. 232, February 2009, I-Tech, Vienna, Austria

1.2 The sonar sensors

Many recent localization strategies rely on accurate sensors, such as laser range scanners (Hähnel et al. 2003; Biber & Straβer 2003; Montesano et al. 2005; Minguez et al. 2006). Today, off the shelf laser sensors provide thousands of readings per second with a sub degree angular resolution. Other sensors, such as standard Polaroid ultrasonic range finders, are only able to provide tenths of readings per second, with angular resolutions one or two orders of magnitude worse than laser. Moreover, effects such as multiple reflections or cross-talking are very frequent in sonar sensing, producing large amounts of readings not corresponding to real objects in the environment. Figure 1 compares the sets of readings gathered simultaneously by a laser scanner and an ultrasonic range finder along the same trajectory. The laser set contains 150 times more readings than the sonar set. This provides a clear idea of the sparse sets of readings provided by ultrasonic range finders when compared to laser scanners. Also, it can be observed how the ultrasonic set contains large amounts of wrong readings, due to the previously mentioned effects.

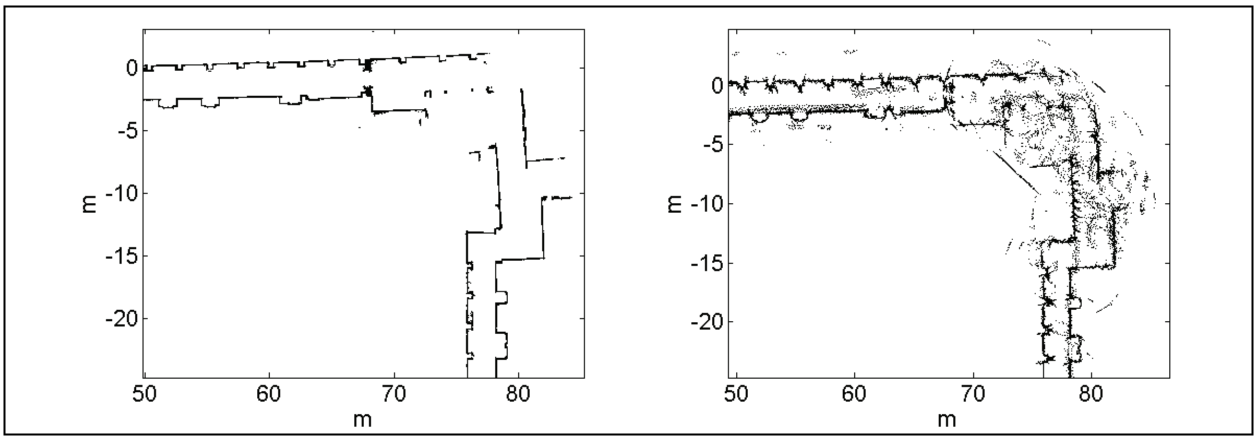


Fig. 1. Example of readings gathered by a laser scanner (left) and a ring of 16 ultrasonic range finders (right).

Nevertheless, ultrasonic range finders have interesting properties that make them appealing in the mobile robotics community (Lee 1996). On the one hand, their size, power consumption and price are better than those of laser scanners. Consequently, they are well suited for low cost and domestic robots, such as automatic vacuum cleaners. On the other hand, their basic behaviour is shared with underwater sonar sensors, which are vastly used in underwater and marine robotics. Thus, typical underwater sonar, although being far more complex than standard Polaroid ultrasonic range finders, can take profit of those localization techniques accounting for sonar limitations. In the context of this work, the terms *sonar*, *ultrasonic range finder* and *ultrasonic sensor* will be used interchangeably, and will refer to standard time-of-flight Polaroid ultrasonic range finders.

The validity of ultrasonic range finders to perform localization has been demonstrated by different researchers. For instance, Tardós et al. (Tardós et al. 2002) use a perceptual grouping technique to identify and localize environmental features, such as lines and corners. These features are correlated using robust data association to perform SLAM with sonar sensors. Also, Groβmann et al. (Groβmann & Poli 2001) confront the sonar localization problem by means of the Hough transform and probability grids to detect walls and corners. Burguera et al. (Burguera et al. 2008a) adopt a different approach, named spIC (*sonar probabilistic Iterative Correspondence*), not requiring environmental features to be detected. They have shown that scan matching localization, even in its most basic expression (Burguera et al. 2005), can be applied to sonar sensors if their limitations and uncertainties are appropriately taken into account. Also, a recent laser scan matching technique, the NDT (*Normal Distributions Transform*) (Biber & Straβer 2003) has been adapted to work with sonar sensors (Burguera et al. 2008b).

1.3 Probabilistic methods

It is broadly accepted that probabilistic methods are the most promising ones to deal with sensor and pose uncertainties in real-time (Thrun et al. 2005). A key concept in probabilistic robotics is that of *belief*. The beliefs, which reflect the robot's internal knowledge about its state, are represented by probability distributions. In the localization context, the beliefs reflect the robot's internal knowledge about its pose in the space. The most general approach to compute beliefs is the *Bayes filter*, which determines the belief distribution from control data and measurements. However, the general Bayes filter algorithm is not a tractable implementation for continuous state spaces, such as the one of robot poses.

Gaussian filters are one of the earlier tractable implementations of Bayes filters. One particular Gaussian filter, the *Kalman filter* (Kalman 1960), has been vastly used to perform mobile robot localization and SLAM. A Kalman filter represents the belief by means of a normal distribution. Mainly due to the normal distribution assumption, Kalman filters fail to represent ambiguities and to recover from localization (Neira & Tardós 2001; Castellanos et al. 2004). These problems are especially relevant when dealing with ultrasonic range finders. On the one hand, the low sonar angular resolution may lead to ambiguous robot pose estimates. On the other hand, wrong readings due to multiple reflections and cross-talking combined with the low measurement rate of ultrasonic range finders may lead the filter to unrecoverable localization failures.

An alternative tractable implementation of the Bayes filter is the *Particle filter* (Metropolis and Ulam 1949, Doucet et al. 2001). In particle filters, the belief distribution is represented by a set of samples, called *particles*, randomly drawn from the belief itself. The particle filter is in charge of recursively updating the particle set. Dellaert et al. (Dellaert et al. 1999) and Fox et al. (Fox et al. 1999) introduced particle filters in the localization context, defining the so-called MCL (*Monte Carlo Localization*). Since then, particle filters have been successfully applied to SLAM (Montemerlo et al. 2002; Hähnel et al. 2003), multi-robot localization (Fox et al. 2000) and localization given an *a priori* map both using laser (Yaqub & Katupitiya 2007) and sonar sensors (Thrun et al. 2001), among many other applications.

Particle filters are nonparametric implementations of Bayes filters. They are said to have two important advantages. First, they can approximate a wide range of probability distributions, even multimodal. When compared to Kalman filters, which can only deal with normal distributions, this feature constitutes an important benefit. These filters are much better suited than Kalman filters to represent ambiguities and to cope with localization failures. The second advantage of particle filters is that, even its most straightforward implementation exhibits very good results when applied to localization. Thus, a particle filter constitutes an excellent tool to perform localization using ultrasonic range finders as exteroceptive sensors.

A key point in a particle filter is the so called *measurement model*. Broadly speaking, the measurement model is in charge of determining how likely the current sensor readings can be explained by each particle. This is usually accomplished by means of an *a priori* map. The current sensor readings are matched against the map and the degree of matching defines the measurement model.

Silver et al. (Silver et al. 2004) proposed a method not requiring any *a priori* map and dealing with underwater sonar sensors. Their proposal was to store a small local history of range readings for each particle, so that the current readings could be matched against each of the local histories. In order to compute the degree of matching, they borrowed a concept from the scan matching community. They compute the degree of matching as a function of the ICP (*Iterative Closest Point*) scan matching error (Lu & Milios 1997). Although this approach was only tested in simulation, it does not require any *a priori* map and exhibits very good results.

1.4 Chapter overview

In this document we describe the use of the ICP scan matching as a measurement model in a particle filter to perform mobile robot localization using standard, terrestrial, Polaroid ultrasonic range finders. The particles are augmented with local environment information. This local information is recursively updated at each time step, allowing the localization process to be performed without any *a priori* map. Also, the aim of this local information is to deal with the sparseness of the sets of sonar readings.

In order to validate and measure the quality of this approach, sonar and laser data has been simultaneously gathered in different environments. Using the laser readings, a ground truth has been constructed. Then, the sonar-based particle localization is evaluated by comparing its results to the ground truth. The presented evaluation method takes into account the whole robot trajectory, instead of only its end points. The experiments evaluate different algorithm's parameters. The experimental results show how the proposed approach to sonar-based localization is able to provide robust and accurate robot pose estimates.

This chapter is structured as follows. The general particle filter operation, as well as the specific details of the sonar-based particle localization, is provided in Section 2. The notation used along the paper is also presented in this section. Section 3 focuses on the measurement model and the introduction of local environment information on the particles. A quantitative evaluation method is presented in Section 4. Also in this section, the experimental results evaluating the algorithm are shown and discussed. Section 5 concludes the chapter, and some proposals of future work are given in Section 6.

2. The particle filter

2.1 Overview and notation

The key idea in mobile robot localization is to estimate the robot pose from sensor data. Thus, in the localization context, the robot pose is the state to be estimated. However, the robot pose is usually not directly observable by sensors. In consequence, the robot pose has to be inferred from sensor data. In order to accomplish this state estimation process, at least two models are necessary. On the one hand, a model describing the evolution of the robot poses with time. On the other hand, a model that relates the sensor measurements to the robot poses. The former is commonly named the *motion model*, although it is also referred to as the *system model* or the *plant model*. The latter is the so called *measurement model*.

As stated previously, Bayes filters constitute a general, widely used, approach to state estimation. They are recursive estimators consisting on two main steps: the *control update* and the *measurement update*. The control update predicts the robot pose one time step forward, by means of the motion model. The measurement update uses the latest sensor readings and the measurement model to modify the prediction. This update is based on the Bayes theorem, and that is why these recursive estimators are named Bayes filters.

The general Bayes filter algorithm involves integration over the whole state space. In localization, the state space is the continuous space of robot poses. In consequence, the general Bayes filter algorithm is not computationally tractable in most of the localization problems. Nevertheless, there exist several tractable implementations of the Bayes filter. Among them the particle filter is capturing the attention of the localization community. The key idea is to represent the belief by a set of weighted random samples called particles. Thanks to this, the filter is able to deal with arbitrary probability distributions, not necessarily unimodal. Thus, they are particularly well suited to perform sonar-based localization, and they will focus the attention of this chapter. Additionally, the presented approach does not require an *a priori* map of the environment. The particular implementation of the particle filter to perform such task is described in detail in Section 2.2. Now, some notation is provided.

Let z_t and u_t denote the sonar measurements and the control vector, respectively, at time step *t*. In the context of this chapter, u_t corresponds to odometry data. Let the set of particles at time step *t* be defined as follows:

$$X_t := \{(x_t^{[m]}, w_t^{[m]}, s_t^{[m]}), 1 \leq m \leq M\} \tag{1}$$

where *M* is the number of particles. Each $x_t^{[m]}$ is a concrete instantiation of the robot pose $[x, y, \theta]^T$ at time *t*. Each $w_t^{[m]}$ is the particle *importance factor*, also referred to as *weight*, so that $w_t^{[m]} \propto p(z_t|x_t^{[m]})$. A key issue in the presented approach is $s_t^{[m]}$, a short history of the most recent *k* sets of sonar readings. This history constitutes the particle *local map*, and it is intended to cope with the low amount of readings provided by ultrasonic range finders. Also, these local maps are recursively updated during the filter operation, letting the localization process work without the use of *a priori* maps. The use and on-line building of $s_t^{[m]}$ is one of the novelties of the presented approach, and will be described in detail in Section 3. In the context of this chapter, the terms *local map*, *history* and *readings history* will be used interchangeably and will refer to $s_t^{[m]}$.

Let $\bar{x}_t^{[m]}$ denote the relative robot motion from time step *t-1* to time step *t* according to the particle *m*. Finally, let the operators $\ominus$ and $\oplus$ denote the inversion and the compounding transformations, similarly to those defined by Smith et al. (Smith et al. 1990). These operators will be now described, together with two additional compounding operators for transforming the references of a point (Tardós et al. 2002) and of a set of points.

Only to perform such description, the following notation will be used. Let $x_B^A = [x_1, y_1, \theta_1]^T$denote the location of a coordinate frame *B* relative to a coordinate frame *A*. Let $x_C^B = [x_2, y_2, \theta_2]^T$be defined similarly. Finally, let $x_p^B = [x_3, y_3]^T$ denote the location of the point *p* relative to the coordinate frame *B*. The compounding $x_C^A = x_B^A \oplus x_C^B$ denotes the location of the coordinate frame *C* relative to *A*, and is computed as follows:

$$x_C^A = x_B^A \oplus x_C^B = \begin{bmatrix} x_1 + x_2 \cos\theta_1 - y_2 \sin\theta_1 \\ y_1 + x_2 \sin\theta_1 + y_2 \cos\theta_1 \\ \theta_1 + \theta_2 \end{bmatrix} \tag{2}$$

The inversion $x_A^B = \ominus x_B^A$ denotes the location of coordinate frame *A* relative to *B* as follows:

$$x_A^B = \ominus x_B^A = \begin{bmatrix} -x_1 \cos\theta_1 - y_1 \sin\theta_1 \\ x_1 \sin\theta_1 - y_1 \cos\theta_1 \\ -\theta_1 \end{bmatrix} \quad (3)$$

The compounding $x_p^A = x_B^A \oplus x_p^B$ denotes the location of the point *p* relative to the coordinate frame *A* as follows:

$$x_p^A = x_B^A \oplus x_p^B = \begin{bmatrix} x_1 + x_3 \cos\theta_1 - y_3 \sin\theta_1 \\ y_1 + x_3 \sin\theta_1 + y_2 \cos\theta_1 \end{bmatrix} \quad (4)$$

Finally, if the right-hand operand of the compounding transformation is a set of points, the transformation is applied individually to each point and, thus, the compounding returns the resulting set of points. Figure 2 summarizes the notation used along this chapter.

2.2 Sonar-based particle localization

A particle filter builds the particle set X_t recursively from the particle set X_{t-1} one time step earlier. Thus, it is necessary to start the recursion by defining the initial particle set X_0. If an *a priori* map is available, this initialization is accomplished by uniformly distributing $x_0^{[1:M]}$ over the free space in the map. However, the presented approach uses local maps to avoid the need for previous information. In consequence, the particle set initialization has to be in charge of building $s_0^{[1:M]}$.

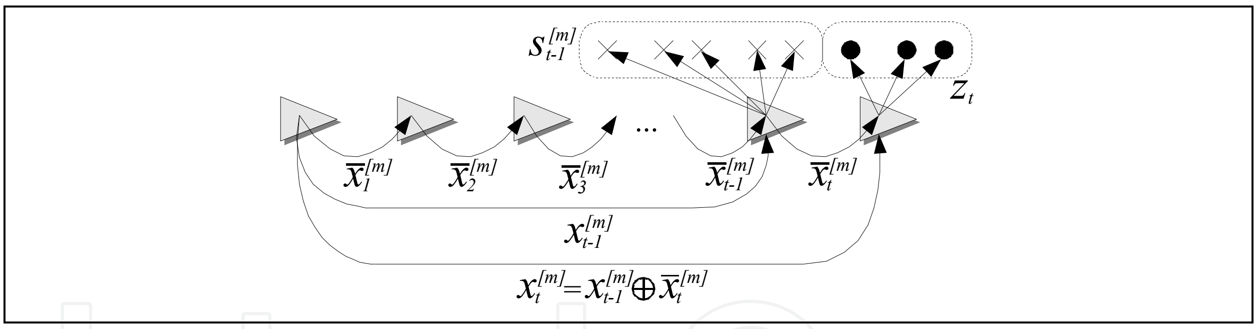


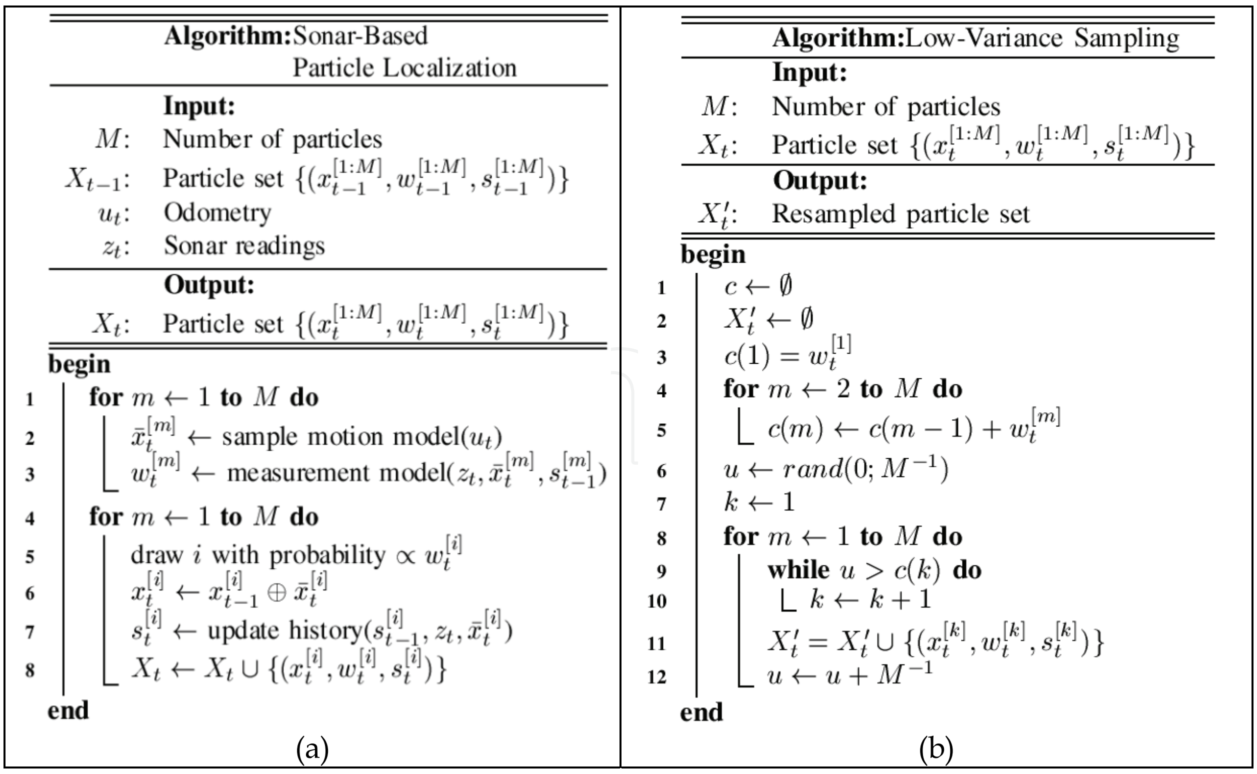
Fig. 2. Notation used for sonar-based particle filtering

To initialize the particle set, the robot has to move during *k* time steps computing its pose using odometry. Then, the robot pose $x_0^{[1:M]}$ for all particles is set to the odometric pose estimate after the mentioned *k* time steps. During this initialization, *k* sets of sonar readings are gathered. Their $[x, y]^T$coordinates are represented with respect to a coordinate frame located in $x_0^{[1:M]}$ using the odometry estimates. The initial local map $s_0^{[1:M]}$ for all particles is then set to the mentioned *k* sets of sonar readings.

Although this initial dependence on odometry may seem problematic, it is not if the value of *k* is appropriately chosen. Different values for this parameter will be tested and experimentally evaluated along this chapter. It will be shown that good values for this parameter are around *k=100*. So, let us assume for the moment this value. Let also assume a mobile robotic platform providing odometric and sonar readings at steps of 100ms. This time step is quite common. In this case, the robot has to rely solely on odometry during the first 10s of operation. In many robot applications, the odometric error accumulated during 10s is negligible if compared to the whole mission execution.

After the initialization process, the sonar-based particle filter localization algorithm is executed at each time step. Figure 3-a shows the algorithm, where two loops involving the whole particle set can be observed. The first loop, from line 1 to line 3, predicts the robot pose by means of the motion model. Thus, it constitutes the control update. The second loop, from line 4 to line 8, updates the particle set according to the sensor readings and constitutes the measurement update. It is important to remark that, although the measurement model is executed in line 3, the weights it computes are not used until line 5. Thus, the measurement update is performed from line 4 to 8 although the measurement model appears in line 3.

Line 2 is in charge of sampling the motion model. As stated previously, in order to perform a state estimation process, a motion model is necessary. In general terms, the motion model describes the evolution of the robot pose with time. Thus, in general, the motion model provides an estimate of where the robot is at time *t*, given its previous pose at time *t-1* and the current control vector u_t. However, our proposal for the motion model is not to compute the absolute robot pose at time *t*, but relative motions from one time step to the next. In consequence, line 2 generates hypothetical robot motions $\bar{x}_t^{[m]}$ from time step *t-1* to time step *t* using a stochastic motion model that does not depend on $x_{t-1}^{[m]}$. This step involves sampling from the distribution $p(\bar{x}_t|u_t)$, where $\bar{x}_t$ represents a robot motion from time step *t-1* to time step *t*. This distribution depends on the specific robot configuration, and it is out of the scope of this chapter to discuss it. Relevant information on this subject can be found on (Thrun et al. 2005).



```
Algorithm: Sonar-Based Particle Localization
Input:
  $M$: Number of particles
  $X_{t-1}$: Particle set $\{(x_{t-1}^{[1:M]}, w_{t-1}^{[1:M]}, s_{t-1}^{[1:M]})\}$
  $u_t$: Odometry
  $z_t$: Sonar readings
Output:
  $X_t$: Particle set $\{(x_t^{[1:M]}, w_t^{[1:M]}, s_t^{[1:M]})\}$
begin
1   for $m \leftarrow 1$ to $M$ do
2       $\bar{x}_t^{[m]} \leftarrow$ sample motion model($u_t$)
3       $w_t^{[m]} \leftarrow$ measurement model($z_t, \bar{x}_t^{[m]}, s_{t-1}^{[m]}$)
4   for $m \leftarrow 1$ to $M$ do
5       draw $i$ with probability $\propto w_t^{[i]}$
6       $x_t^{[i]} \leftarrow x_{t-1}^{[i]} \oplus \bar{x}_t^{[i]}$
7       $s_t^{[i]} \leftarrow$ update history($s_{t-1}^{[i]}, z_t, \bar{x}_t^{[i]}$)
8       $X_t \leftarrow X_t \cup \{(x_t^{[i]}, w_t^{[i]}, s_t^{[i]})\}$
end
```

(a)

```
Algorithm: Low-Variance Sampling
Input:
  $M$: Number of particles
  $X_t$: Particle set $\{(x_t^{[1:M]}, w_t^{[1:M]}, s_t^{[1:M]})\}$
Output:
  $X_t'$: Resampled particle set
begin
1   $c \leftarrow \emptyset$
2   $X_t' \leftarrow \emptyset$
3   $c(1) = w_t^{[1]}$
4   for $m \leftarrow 2$ to $M$ do
5       $c(m) \leftarrow c(m-1) + w_t^{[m]}$
6   $u \leftarrow rand(0; M^{-1})$
7   $k \leftarrow 1$
8   for $m \leftarrow 1$ to $M$ do
9       while $u > c(k)$ do
10          $k \leftarrow k + 1$
11      $X_t' = X_t' \cup \{(x_t^{[k]}, w_t^{[k]}, s_t^{[k]})\}$
12      $u \leftarrow u + M^{-1}$
end
```

(b)

Fig. 3. Sonar-Based Particle Localization algorithm (a) and Low Variance Sampling algorithm (b).

The second model necessary to perform the state estimation is the measurement model, which relates the sonar readings to the robot pose. Line 3 uses the measurement model to incorporate the current readings z_t into the particle set by computing the importance factor $w_t^{[m]}$. Those particles with the relative motions that better explain the current readings will have better weights. This is accomplished by correlating z_t and $s_{t-1}^{[m]}$ by means of an ICP-like approach and will be described in Section 3.

Line 5 executes the so called *resampling*, also referred to as *importance sampling*. The importance sampling is the core of the measurement update step. At this point, the algorithm draws with replacement *M* particles. The probability of drawing each particle is proportional to its importance factor. Differently speaking, during the importance sampling, those particles with better weights have higher probability to remain in the particle set.

There is a problem in particle filters directly related to the importance sampling. The statistics extracted from the particles may differ from the statistic of the original density, because the particle set only holds a finite number of random samples. This problem may lead to a *degeneracy phenomenon* through repetitive resampling (Sanjeev Arulampalam et al. 2002). Degeneracy appears when, after a number of resampling steps, all but one particle have negligible weights. Among the existing resampling strategies, the *low variance sampling* has proved to be very efficient, in computational terms, while reducing the degeneracy phenomenon. The underlying idea is to select the samples in a sequential stochastic process instead of independently. A comprehensible description of the algorithm is available in (Thrun et al. 2005). Because of the mentioned advantages, the low variance sampling has been adopted in the present work. The algorithm is presented in Figure 3-b.

Going back to Figure 3-a, the line 6 is in charge of updating the global robot pose for each particle selected during the resampling step. This is accomplished by compounding the global robot pose at time *t-1* with the relative motion from time step *t-1* to time step *t*. This idea is illustrated in Figure 2.

Line 7 is in charge of building the new local map of each particle, by adding the current set of sonar readings and discarding to oldest readings so that the map size remains constant along the whole mission execution. This process will be described in Section 3.

Finally, line 8 constructs the new particle set X_t. After this step, depending on the specific robot application, the particle set may be treated in different ways. For instance, some applications need a single vector $[x, y, \theta]^T$ informing the most likely robot pose. In that cases, the mean of $x_t^{[1:M]}$ may be used. Some other applications require a continuous probability density function to be extracted from the samples. In those cases, techniques such as *Gaussian approximation*, *K-Means* or *Kernel Density Estimation* can be used. The reader is directed to (Thrun et al. 2005) to learn more about these density extraction techniques.

3. Matching sets of readings

3.1 Overview and notation

In particle filters, the measurement model is in charge of computing the weights of the particles. In particle filter localization, the weights represent the likelihood of having the current set of readings z_t at the robot pose $x_t^{[m]}$. Thus, $w_t^{[m]} \propto p(z_t|x_t^{[m]})$. This dependence on the absolute robot pose is useful if an *a priori* map is available, because the range readings can be matched against the global map using the absolute robot pose.

However, one advantage of the presented approach is that it does not require a previously constructed map. Instead, local maps are recursively built during the mission execution. For a given particle, the local map $s_{t-1}^{[m]}$ is represented with respect to the coordinate frame in $x_{t-1}^{[m]}$ (see Figure 2). Also, the presented motion model generates $\bar{x}_t^{[m]}$, the relative motions from time step *t-1* to time step *t*. Taking into account that the current set of readings z_t has been gathered at time *t*, the particle weight can be computed by evaluating the degree of matching between $\bar{x}_t^{[m]} \oplus z_t$ and $x_{t-1}^{[m]}$. Figure 2 clarifies this point. Thus, in our approach, $w_t^{[m]} \propto p(z_t|\bar{x}_t^{[m]}, s_{t-1}^{[m]})$. Broadly speaking, the idea is to weight the particles according to the existing overlap between the current set of readings and the stored maps. Computing the overlap between two sets of range readings is a common practice in the scan matching community. Thus, some scan matching concepts will be used in this section. Next, some notation is introduced.

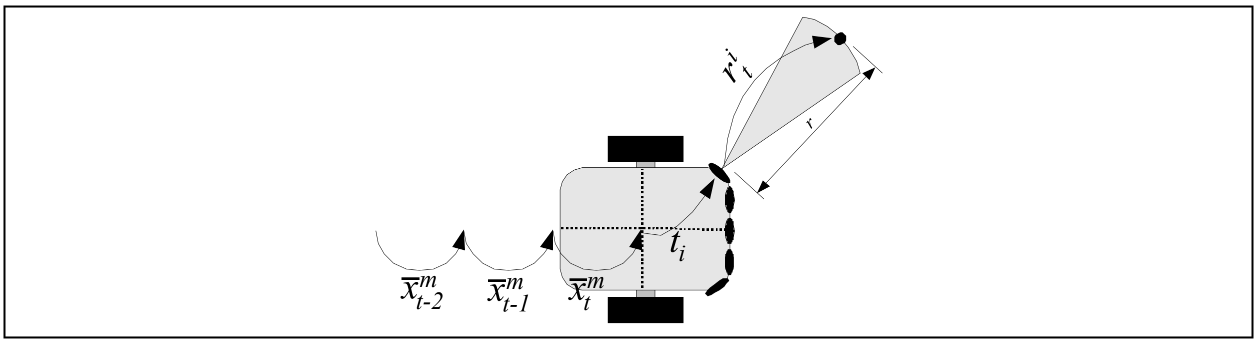


Fig. 4. Relations between the coordinate frames used by the measurement model. The circular sector represents the sonar beam. The dashed cross is the robot coordinate frame.

Let r_t^i represent the range reading provided by the *i-th* sonar sensor at time step *t*. Let this reading be represented with respect to a coordinate frame located on the sonar sensor and aligned with the ultrasonic acoustic axis. Thus, r_t^i has the form $[r, 0]^T$, where *r* is the raw range provided by the sensor.

Let t_i denote the relative position of the sonar sensor *i* with respect to the robot reference frame. Ultrasonic range finders are assumed to be at fixed positions on the robot body. Consequently, t_i does not change over time. That is why the subindex *t* has been dropped. Figure 4 illustrates the notation.

3.2 Building the local maps

At time *t*, the array of ultrasonic range sensors provides a set of raw range readings. The set z_t is built from the raw range readings as follows:

$$z_t = \{t_i \oplus r_t^i, \forall i \in VS_t\} \tag{5}$$

where VS_t is the set of sonar sensors that have generated a reading during the time step *t*. Each item in z_t will be denoted z_t^i, meaning that it was gathered at time *t* and produced by the *i-th* sonar sensor.

Let S_{new} be defined as the set of readings in z_t represented with respect to the coordinate frame of $x_{t-1}^{[m]}$ using the relative motion $\bar{x}_t^{[m]}$ proposed by the particle:

$$S_{new} = \bar{x}_t^{[m]} \oplus z_t = \{\bar{x}_t^{[m]} \oplus z_t^i, \forall i \in VS_t\} \tag{6}$$

Each item in S_{new} will be denoted by p_t^i, meaning that it has been generated from z_t^i.
To ease notation, let S_{old} denote the local map $s_{t-1}^{[m]}$. It was stated previously that all the readings in S_{old} are represented with respect to the coordinate frame of $x_{t-1}^{[m]}$. This is accomplished by building S_{old} as follows:

$$\begin{aligned} S_{old} = \quad & \{z_{t-1}, \\ & \ominus\bar{x}_{t-1}^{[m]} \oplus z_{t-2}, \\ & \ominus\bar{x}_{t-1}^{[m]} \oplus (\ominus\bar{x}_{t-2}^{[m]}) \oplus z_{t-3}, \ldots, \\ & \ominus\bar{x}_{t-1}^{[m]} \oplus \ldots \oplus (\ominus\bar{x}_{t-k+1}^{[m]}) \oplus z_{t-k}\} \end{aligned} \tag{7}$$

where *k* is the local map size. By observing the previous equation it is easy to see that $s_t^{[m]}$ can be obtained recursively from $s_{t-1}^{[m]}$, z_t and $\bar{x}_t^{[m]}$. This recursive update, which is performed by the update history function in line 7 of Figure 3-a, can be expressed as follows:

$$s_t^{[m]} = ((\ominus\bar{x}_t^{[m]}) \oplus s_{t-1}^{[m]}) \cup z_t \tag{8}$$

First, the readings in $s_{t-1}^{[m]}$ are represented with respect to the coordinate frame of $x_t^{[m]}$ by compounding them with $\ominus\bar{x}_t^{[m]}$. Then, the new set of readings z_t is added. Finally, although not been represented in Equation (8), the oldest readings in the resulting set have to be deleted so that the size of the local maps remains constant along the whole mission execution.

3.3 The measurement model

There exist many algorithms to match sets of range readings in the scan matching literature] (Lu & Milios 1997; Rusinkiewicz & Levoy 2001; Pfister et al. 2004; Burguera et al. 2008a). Most of them follow the structure proposed by the ICP algorithm. The key step in the ICP algorithm is the establishment of point to point correspondences between readings in two consecutive range scans. These correspondences are established by means of the Euclidian distance, and they give information about the degree of matching between two sets of readings. Our proposal is to measure the degree of matching between S_{new} and S_{old} in that way. This will constitute our measurement model.
Let p_i and q_j be points in S_{new} and S_{old} respectively. To decide whether a correspondence between p_i and q_j can be established or not, the Euclidian distance is used:

$$d(p_i, q_j) = \sqrt{(p_i - q_j)^T (p_i - q_j)} \tag{9}$$

For each $p_i \in S_{new}$, the closest point $q_j \in S_{old}$ according to the distance in Equation (9) is selected to be the corresponding point. Thus, the set *C* of correspondences is defined as follows:

$$C = \{(p_i, q_j), \forall p_i \in S_{new} | q_j \in S_{old}, q_j = \arg\min(d(p_i, q_j))\} \tag{10}$$

Broadly speaking, the idea is to establish correspondences between the points in S_{new} and S_{old} that are closer in the Euclidian sense. This is commonly referred to as the *closest point rule*.

The sum of Euclidian distances between pairs of corresponding points is a good indicator of the degree of matching between S_{new} and S_{old}: the worse the matching, the bigger the sum of distances. However, the importance factor represents the opposite idea: particles that produce better matching should have higher weights. In consequence, the importance factor for a particle *m* is computed as follows:

$$w_t^{[m]} = (\sum_{(p_i,q_j)\in C} d(p_i,q_j))^{-1} \tag{11}$$

In order to avoid numerical problems, those situations where the sum of distances is close to zero should be especially taken into account. However, experimental results suggest that, due to the noisy nature of sonar sensors, these situations are extremely unusual.

4. Experimental results

4.1 Experimental setup

In order to evaluate the presented approach, a Pioneer 3-DX robot, endowed with 16 Polaroid ultrasonic range finders and a Hokuyo URG-04LX laser scanner, has been used. The robot has moved in four different environments in our university, gathering various data sets. Each data set contains the odometry information, the sonar range readings and the laser range readings. The laser readings have only been used to obtain ground truth pose estimates. In order to obtain such ground truth, the ICP scan matching algorithm has been applied to the laser readings. Then, the wheel encoder readings have been corrupted with Gaussian noise ($\mu = 0m/s$ and $\sigma^2 = 0.0025$) to simulate worse floor conditions. Thus, the quality of our algorithm operating with noisy and sparse sets of sonar readings in bad floor conditions is compared to a well known localization algorithm operating with dense and high quality laser readings and good floor conditions.

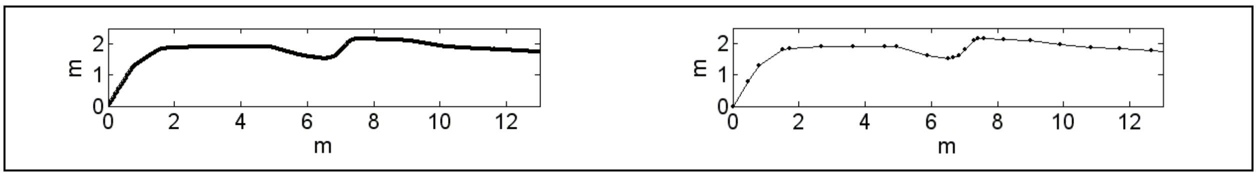


Fig. 5. Fragment of a real trajectory (left) and the polyline that approximates it (right). The dots represent the vertexes.

In order to quantitatively compare odometry and the different particle filter configurations, the following procedure has been used. First, the trajectories obtained by odometry, particle filter and ground truth are approximated by polylines. The vertex density of each polyline increases in those regions with significant amount of robot rotation. Also, the maximum robot motion between two vertexes has been set to 1m. This kind of approximation is useful to overcome the local perturbations in the individual motion estimates, both for odometry, particle filter and ground truth. Figure 5 exemplifies the polyline approximation. Then, the individual edges of the trajectory being evaluated are locally compared to those of the ground truth. The Euclidian distance between their end points is used as a measure of the edge error. Finally, the edge errors for the trajectory being evaluated are summed. This sum is normalized, using the path lengths between vertexes and the number of edges, and constitutes the *trajectory error*. Due to the mentioned normalization, the errors of different trajectories can be compared. It is important to remark that, as a result of the mentioned procedure, the evaluation takes into account the whole trajectory, not only its end points.

Two different experiments have been performed. The first experiment evaluates our approach with respect to the number of particles, *M*. The second experiment evaluates our approach with respect to the local map size, *k*.

4.2 Evaluating the influence of the number of particles

The first experiment evaluates the quality and the execution time of our approach with respect to the number of particles. The values of *M* that have been tested are 10 and 50, to observe how the algorithm behaves with a low number of particles, and then 100, 200 and 400 particles. The local map sizes has been set to *k=100*. The trajectory error has been computed for odometry and particle filter using the mentioned number of particles.

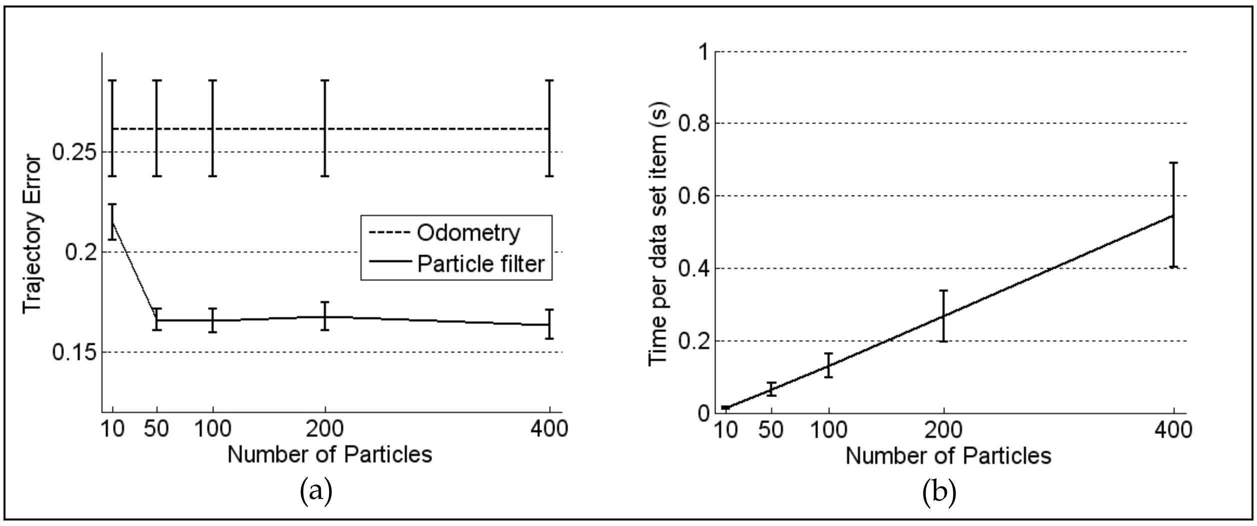


Fig. 6. Experimental results obtained using different numbers of particles and setting the history size to *k=100*. (a) Means and standard deviations of the trajectory errors. (b) Means and standard deviations of the execution time per data set item on a Matlab implementation.

Figure 6-a depicts the mean and the standard deviation of the obtained trajectory errors for all data sets. The graphical representation of the standard deviation has been reduced to a 20% to provide a clear representation, both for odometry and particle filter. Also, although the odometric error does not depend on the number of particles, it has been included on the figure for comparison purposes.

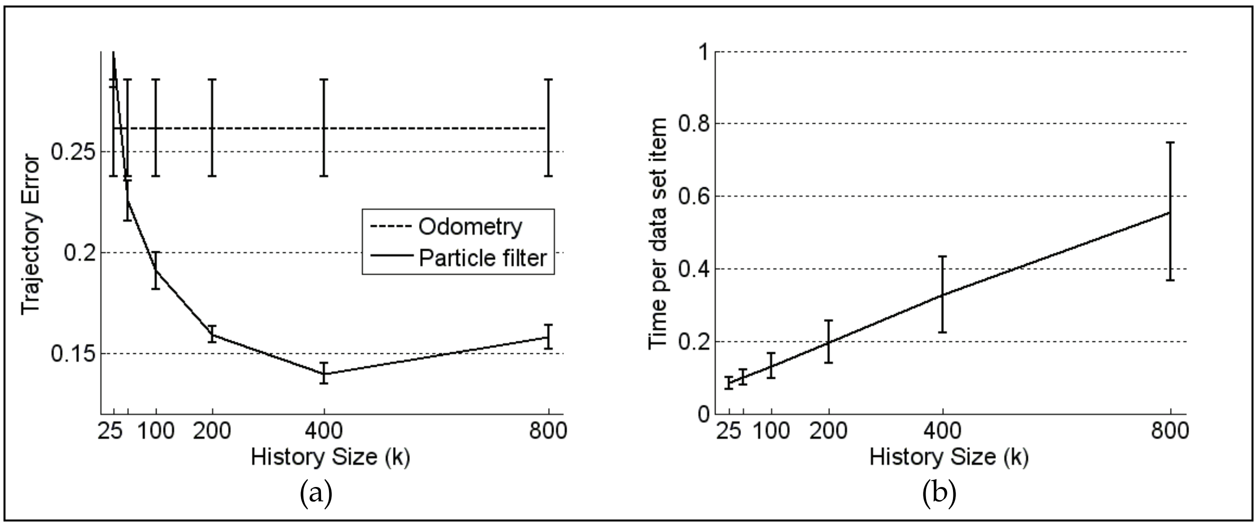
The first thing to be noticed is that the presented approach is able to reduce the odometric error in all cases. Even if only 10 particles are used, the resulting trajectory is, in mean, a 21.9% better than odometry. In the case of 400 particles, the resulting trajectory achieves, in mean, a 60% of improvement with respect to odometry. Also, the standard deviations of the particle filter errors are significantly lower than those of odometry. This suggests that the quality of the particle filter estimates is barely influenced by the initial odometric error.

The second thing to be noticed is that a large error reduction appears from 10 to 50 particles. From this number of particles onward, the error reduction is very small. This suggests that the behaviour of our algorithm does not strongly depend on the number of particles. It also suggests that using a number of particles between 50 and 100 would be a good choice, more if the execution times are taken into account.

Figure 6-b shows the mean and the standard deviation of the execution times per data set item, with respect to the number of particles. It is important to remark that these execution times correspond to a non optimized Matlab implementation. Thus, the absolute values are meaningless as a C++ implementation will greatly increase the execution speed. The interest of these results is that the execution time is strongly linear with the number of particles. This linear relation reinforces the idea that using between 50 and 100 particles is the better choice: the small improvement of using more particles does not compensate the increase in computation time.

4.3 Evaluating the influence of the local maps size

The second experiment evaluates the quality and the execution time of our approach with respect to the local maps size. Now, the number of particles is set to 100, as it has shown to be a good choice, and the history sizes k=25, k=50, k=100, k=200, k=400 and k=800 are tested.



(a) (b)

Fig. 7. Experimental results obtained using different local map sizes and setting the number of particles to *M=100*. (a) Means and standard deviations of the trajectory errors. (b) Means and standard deviations of the execution time per data set item on a Matlab implementation.

Figure 7-a shows the mean and the standard deviation of the trajectory errors, both for odometry and particle filter. The standard deviation has been graphically reduced to a 20% to provide a clear representation.

It can be observed how the effects of the history size are more noticeable than those of the number of particles. For example, if the very short history *k=25* is used, the resulting trajectory is worse than the one provided by odometry. The reason of this problem is that, using a very short history, the influence of spurious and wrong readings in the measurement model is not negligible. Also, it is clear that increasing the history size may lead to better results than increasing the number of particles. For instance, the trajectory obtained using *M=100* and *k=400* is an 87% better than the odometric one, while the trajectory obtained using *M=400* and *k=100* is only a 60% better.

It is important to remark that the quality of the particle filter slightly decreases for *k=800*. This quality reduction is mainly due to the initialization process. As stated previously, the time spent to build the initial particle set X_0 depends on the value of *k*. In our implementation, setting *k=800* means that the robot has to solely rely on odometry during 1 minute and 20 seconds at the beginning of its operation. This dependence on odometry is responsible of the mentioned quality reduction.

Figure 7-b shows the mean and the standard deviation of the execution times per data set item. As in the previous experiment, these times correspond to a non optimized Matlab implementation. Thus, the interest of the execution times does not reside on their absolute values but on their evolution with respect to the history size.

Similarly to the previous experiment, the execution time is strongly linear with the history size. Looking at the Figures 6-b and 7-b , it is clear that taking into account the time consumption, the better choice is to increase the history size rather than the number of particles. For instance, the errors for *M=400* and *k=100* are similar to those of *M=100* and *k=200*, but the mean execution time for the former is more than twice the execution time of the latter.

4.4 Qualitative evaluation

In order to provide a clear understanding of the results, some images are provided for visual inspection. Different trajectories have been plotted, as well as the sonar readings according to each trajectory.

Figure 8 visually depicts some of the results of the first experiment. The quality of the algorithm with respect to the number of particles can be observed. The first row shows the initial odometry estimates in four different environments. The second, third and fourth rows depict the results using an increasing number of particles (10, 100 and 400). All of them correspond to a history size of *k=100*. Finally, the fifth row shows the results of applying ICP to the laser readings. It is important to remark that, although ground truth trajectory has been obtained by matching laser range readings, the visual map shown in the last row has been plotted with the sonar readings to make the visual comparison easier.

It can be observed how, as the number of particles increases, the resulting trajectory becomes more similar to the ground truth. Even in the large environment of the fourth column, where the robot has moved more than 150m, the final pose estimate is very close to the ground truth. The environment in the third column deserves special attention. By observing the initial odometric estimate, it is easy to see that a significant error appears at the beginning of the trajectory. Because the initial particle set X_0 construction requires for the robot to be confident on odometry at the beginning of its operation, this initial error can not be fully corrected. That is why the particle filter provides a visual map rotated with respect to the ground truth. However, the shape of the trajectory is almost identical to the one of the ground truth.

The Figure 9 visually depicts some of the results of the second experiment. The quality of the algorithm with respect to the history size can be observed. The first and fifth rows, which correspond to the initial odometric estimates and the ground truth respectively, are the same that in Figure 8, and are plotted here again to provide a clear idea of the evolution of the pose estimates. The second, third and fourth row correspond to history sizes of *k=25*, *k=50* and *k=200*. In all of them, the number of particles used is *M=100*. Thus, the results for *k=100* can be observed in the third row of Figure 8.

It can be observed how the changes in the history size are clearly reflected in the quality of the resulting trajectory. Very accurate trajectories appear when a history size of 200 is used. As stated previously, the last row corresponds to the localization results of the well known

ICP algorithm applied to accurate and dense sets of laser range readings. On the contrary, our algorithm operates with the sparse and noisy sets of readings provided by standard Polaroid ultrasonic range finders. Moreover, our algorithm operated on a corrupted odometry, simulating bad floor conditions. Thus, it is remarkable that the presented approach is able to provide localization results close to the ones provided by a standard laser scan matching algorithm.

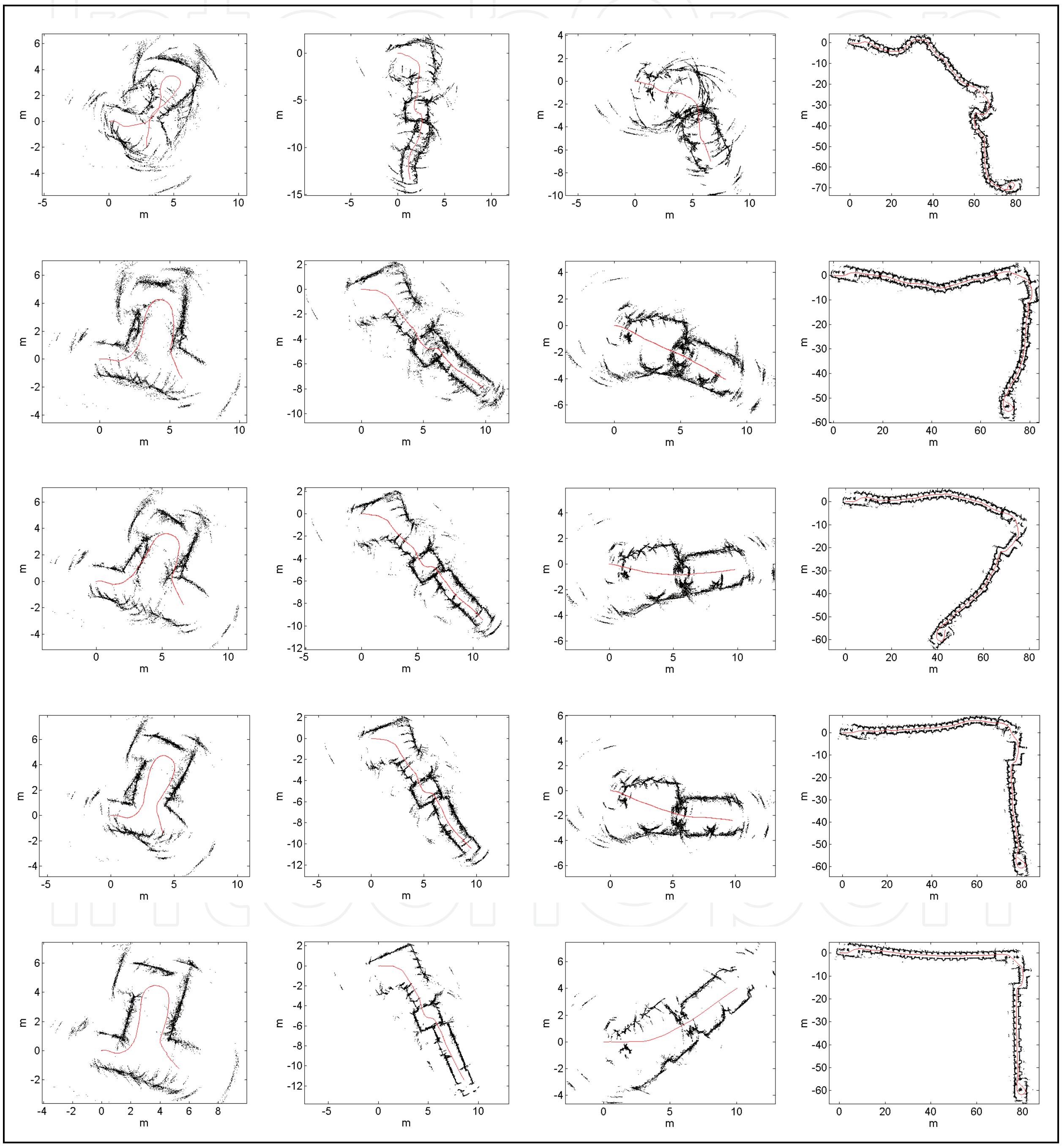


Fig. 8. Trajectories and sonar readings according to odometry (first row), particle filter using 10, 100 and 400 particles respectively (second to fourth row) and ICP laser scan matching (fifth row). The local map size used is *k=100*.

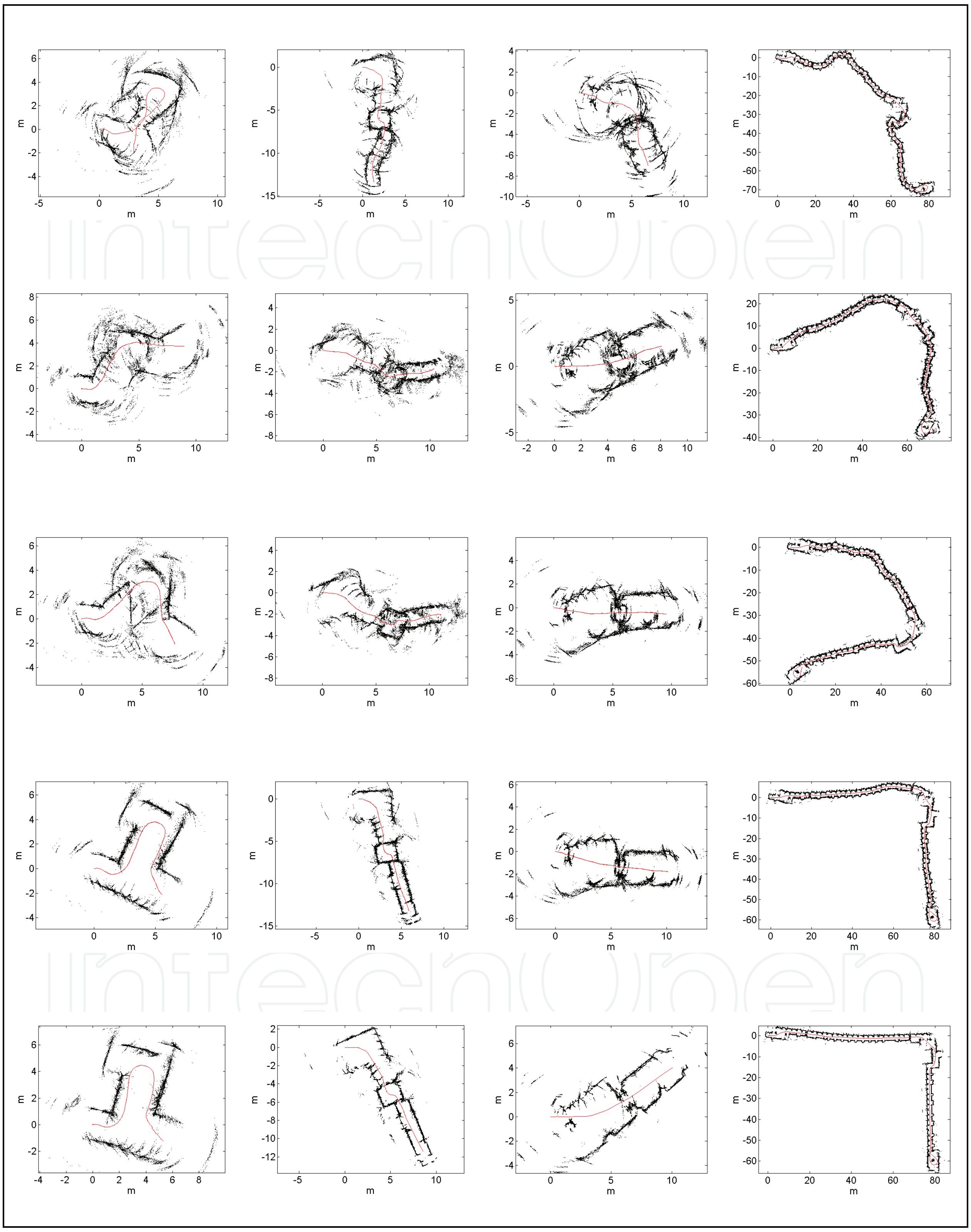


Fig. 9. Trajectories and sonar readings according to odometry (first row), particle filter using history sizes of *k=25*, *k=50* and *k=200* respectively (second to fourth row) and ICP laser scan matching (fifth row). The number of particles used is *M=100*.

5. Conclusion

Localization is a key issue in mobile robotics nowadays. Nearly all robotic tasks require some knowledge of the robot location in the environment. A common way to perform localization is to correlate exteroceptive sensor data at subsequent robot poses. This approach is strongly dependant on the exteroceptive sensor quality. Because of this, many localization algorithms rely on accurate laser range finders, providing dense sets of readings.

Standard ultrasonic range finders are not able to provide such dense and accurate information. That is why they are not frequently used in terrestrial mobile robot localization. However, they are appealing in terms of size, prize and power consumption. Moreover, their basic behaviour is shared with underwater sonar, which is extensively used in underwater and marine robotics. Consequently, a localization technique involving ultrasonic range finders is of great interest in the mobile robotics community.

In this chapter, particle filters have been proposed as a tool to perform localization using ultrasonic range finders. One of the advantages of the presented approach is that it does not require the use of previously constructed maps. Thus, it is suitable even for environments where no *a priori* knowledge is available. This is accomplished by recursively building local maps, which represent the local view that each particle in the filter has about the surrounding environment. Being the local map size constant, the time consumption required to deal with them is also constant.

The measurement model, which is in charge of computing the weights for the particles, has been defined similarly to the closest point rule of the ICP scan matching algorithm. The idea for the measurement model is to use the closest point rule to decide the amount of existing overlap between the current set of sonar readings and each of the local maps.

An experimental setup, involving the construction of a ground truth using accurate and dense laser readings, has been presented. Also, a technique to quantitatively compare different trajectories is discussed. By comparing different particle filter configurations with the ground truth, numerical error measures are obtained.

Two experiments have been defined. The first evaluates the effects of different sizes for the particle set. The second measures the effects of different sizes for the local maps. In both experiments, both the quality of the estimates and the time consumption has been observed. The results suggest that, thanks to the use of particle filters high quality localization results can be obtained using standard Polaroid ultrasonic range finders. These results are comparable to those obtained by standard scan matching algorithms applied to laser readings.

6. Future work

The presented measurement model is based on the ICP scan matching algorithm. This algorithm, which has been vastly used by the localization community, has also proved to be effective when applied to sonar readings (Burguera et al. 2005). However, recent works show that other matching approaches are able to provide more accurate and robust estimates (Burguera et al. 2008a; Burguera et al. 2008b). In consequence, it is reasonable to assume that the presented particle filter approach could benefit of these recent matching techniques in the measurement model.

7. Acknowledgment

This work is partially supported by DPI 2005-09001-C03-02 and FEDER funding.

8. References

Biber, P. & Straβer W. (2003). The Normal Distribution Transform: a new approach to laser scan matching, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2743-2748, October 2003.

Burguera, A.; González Y., & Oliver G. (2008a). A probabilistic framework for sonar scan matching localization. *Advanced Robotics*, Vol. 22, No. 11, August 2008, pp. 1223-1241, ISSN: 0169-1864

Burguera, A.; González Y., & Oliver G. (2008b). Sonar scan matching by filtering scans using grids of normal distributions. In *Intelligent Autonomous Systems 10*, Burgard, W; Dillman, R.; Plagemann, C. & Vahrenkamp, N. (Ed.), pp. 64-73, IOS Press, ISBN 978-1-58603-887-8, Netherlands

Burguera, A.; Oliver, G. & Tardós, J.D. (2005). Robust scan matching localization using ultrasonic range finders, *Proceedings of the Conference on Intelligent Robots and Systems (IROS)*, pp. 1451-1456, August 2005, Edmonton (Canada)

Castellanos, J.; Neira, J. & Tardós J.D. (2004). Limits to the consistency of EKF-based SLAM, *Proceedings of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, July 2004

Dellaert, F.; Burgard W. & Thrun, S. (1999). Monte Carlo Localization for mobile robots, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1999

Doucet, A.; de Freitas, J. & Gordon, N.E. (2001). *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, 2001

Fox, D.; Burgard, W.; Dellaert, F. & Thrun, S. (1999). Monte carlo localization: Efficient position estimation for mobile robots, *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 1999, Orlando (USA)

Fox, D.; Burgard W.; Kruppa, H. & Thrun, S. (2000). A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots*, Vol. 8, No. 3, 2000.

Groβmann, A. & Poli, R. (2001). Robust mobile robot localisation from sparse and noisy proximity readings using Hough transform and probability grids. *Robotics and Autonomous Systems*, No. 37, 2001, pp. 1-18.

Hähnel, D.; Burgard, W., Fox, D. & Thrun, S. (2003). An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 206-211, October 2003.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME. Journal of Basic Engineering,* Vol. 82, March 1960, pp. 35-45.

Lee, D. (1996). The Map-Building and Exploration Strategies of a Simple Sonar-Equipped Mobile Robot, Cambridge University Press, 1996.

Lu, F. & Milios, E. (1997). Robot pose estimation in unknown environments by matching 2D range scans. *Intelligent and Robotic Systems*. Vol. 18, No. 3, March 1997, pp. 249-275.

Metropolis, N. & Ulam, S. (1949). The Monte Carlo method. *Journal of the American Statistical Association*, No. 44, 1949, pp. 335-341.

Minguez, J.; Montesano, L. & Lamiraux, F. (2006). Metric-based iterative closest point scan matching for sensor displacement estimation, *IEEE Transactions on Robotics*, Vol. 22, No. 5, October 2006, pp. 1047-1054.

Montemerlo, M.; Thrun, S.; Koller, D. & Wegbreit, B. (2002). FastSLAM: A factored solution to the simultaneous localization and mapping problem, *Proceedings of the AAAI National Conference on Artificial Intelligence*, 2002.

Montesano, L.; Minguez, J. & Montano, L. (2005). Probabilistic scan matching for motion estimation in unstructured environments, *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, pp. 1445-1450, August 2005, Edmonton (Canada)

Neira, J. & Tardós, J.D. (2001). Data association in stochastic mapping using the joint compatibility test, *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 6, 2001, pp. 890-897.

Pfister, S. T.; Kriechbaum, K.L.; Roumeliotis, S. I. & Burdick, J. W. (2004). Weighted range sensor matching algorithms for mobile robot displacement estimation, *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1667-1674, May 2004.

Rusinkiewicz, S. & Levoy, M. (2001). Efficient variants of the ICP algorithm, *Proceedings of the International Conference on 3D Digital Imaging and Modeling (3DIM)*, 2001.

Sanjeev Arulampalam, M.; Maskell, S.; Gordon, N. & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, Vol. 50, No. 2, February 2002, pp. 174-188.

Silver, D.; Bradley, D. & Thayer, S. (2004). Scan matching in flooded subterranean voids, *Proceedings of the IEEE Conference on Robotics, Automation and Mechatronics*, Singapore, December 2004.

Smith, R.; Self M., & Cheeseman P. (1990). Estimating uncertain spatial relationships in robotics. *Autonomous Robot Vehicles*, 1990, pp. 167-193, ISBN: 0-387-97240-4

Tardós, J. D.; Neira, J.; Newman, P.M. & Leonard, J.J. (2002). Robust mapping and localization in indoor environments using sonar data. *International Journal of Robotics Research*, Vol. 21, No. 4, April 2002, pp. 311-330.

Thrun, S.; Burgard, W. & Fox, D. (2005). *Probabilistic Robotics*. The MIT Press. ISBN 0-262-20162-3.

Thrun, S.; Fox, D. Burgard, W. & Dellaert, F. (2001). Robust monte carlo localization for mobile robots. *Artificial Intelligence*, Vol. 128, No. 1-2, 2001, pp. 99-141.

Yaqub, T. & Katupitiya, J. (2007). Laser scan matching for measurement update in a particle filter, *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics,* pp. 1-6, September 2007.

Advances in Sonar Technology
Edited by Sergio Rui Silva

ISBN 978-3-902613-48-6
Hard cover, 450 pages
Publisher I-Tech Education and Publishing
Published online 01, February, 2009
Published in print edition February, 2009

The demand to explore the largest and also one of the richest parts of our planet, the advances in signal processing promoted by an exponential growth in computation power and a thorough study of sound propagation in the underwater realm, have lead to remarkable advances in sonar technology in the last years.The work on hand is a sum of knowledge of several authors who contributed in various aspects of sonar technology. This book intends to give a broad overview of the advances in sonar technology of the last years that resulted from the research effort of the authors in both sonar systems and their applications. It is intended for scientist and engineers from a variety of backgrounds and even those that never had contact with sonar technology before will find an easy introduction with the topics and principles exposed here.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821