

An Improved Particle Filter for Sparse Environments

Edson Prestes, Marcus Ritt, and Gustavo Führ

Instituto de Informática-UFRGS, P.O. Box 15064, 91501-970 Porto Alegre, RS-Brazil

Abstract

In this paper, we combine a path planner based on Boundary Value Problems (BVP) and Monte Carlo Localization (MCL) to solve the wake-up robot problem in a sparse environment. This problem is difficult since large regions of sparse environments do not provide relevant information for the robot to recover its pose. We propose a novel method that distributes particle poses only in relevant parts of the environment and lead the robot along these regions using the numeric solution of a BVP. Several experiments show that the improved method leads to a better initial particle distribution and a better convergence of the localization process.

Keywords: boundary value problems, autonomous navigation, environment exploration, global localization, Monte Carlo localization.

1. INTRODUCTION

The localization process is an essential component of any mobile robotic system during a navigation task in the environment. It endows the robot with the capacity to determine its correct pose using the collected sensory data. This capacity is considered as a prerequisite to make a robot really autonomous. One of the basic problems in localization is *position tracking* that corresponds to a local localization problem and aims to estimate actively the robot's pose (x, y, θ) from a known initial pose. It tries to compensate for noise in the sensors and errors in odometric readings that can arise in a systematic or non-systematic way to determine the most likely world pose.

Local localization techniques are particularly important during the exploration and building of environment maps. They enable the generation of a faithful representation of the environment while minimizing or eliminating the mapping distortion produced by odometric errors. The task of mapping and simultaneously self-localizing is difficult, because the robot needs to determine its pose based on a map that is being currently built and to map precisely the environment using the information on its

pose. This problem is called Simultaneous Localization and Mapping (SLAM) and has attracted the attention of several researchers during last years [20, 3, 2, 4].

Besides the position tracking problem, there are other challenging localization problems like the *kidnapped robot problem* and the *wake-up robot problem*. They comprise a class of global localization problems. In the wake-up problem, the robot must determine its pose without prior knowledge about its initial location. In the kidnapped robot problem, a well-localized robot is carried to another position during its navigation without being told. In both problems, global localization techniques must be able to handle multiple candidate poses at same time.

Most of the methods used for local localization are unable to solve global localization problems. The main reason is that global localization demands to track and to analyze multiple hypotheses simultaneously. For instance, methods for position tracking based on Kalman filtering [23, 17] represent the robot's estimated pose using an unimodal Gaussian distribution, where the distribution mode provides the current robot position and the variance is the accuracy of the estimation. This distribution limits the use of Kalman filters when there are several candidate poses. To circumvent this, Jensfelt and Kristensen [14] propose a probabilistic approach that uses multiple hypotheses to represent the robot pose. These hypothesis are updated according to the observation made by robot sensors. The hypothesis corresponding to the true robot pose will be more distinguishable than false hypotheses.

The most common approach to handle both global and local localization is the particle filter [27]. The algorithms based on particles belong to a class of methods known as Monte Carlo Localization (MCL). MCL algorithms are easily implemented and computationally efficient since they focus the search on certain regions in the space with high likelihood to contain the real robot pose. There are lots of variations of the basic algorithm. For instance, Milstein et al. [19] suggest the use of clusters of particles to handle symmetric environments. They claim that the basic algorithm converges quickly to a single high prob-

ability pose even if there is more than two similar likely poses. Kwon et al. [16] minimize the number of samples used in MCL using topological information about the environment. This information aims to determine where the particles should be placed in the environment. In this case, the particles are drawn around the neighborhood of the topological edges reducing the number of particles and augmenting the algorithms' efficiency. Gasparri et al. [12] propose an evolutionary approach based on genetic algorithms with a clustering method to handle the global localization problem. At each resampling step, the robot executes a dynamical clustering and at each time step, it executes an evolutionary action to find out the local maxima within each cluster.

In this paper, we propose a strategy that combines path planning based on boundary value problems (BVP) and MCL to solve the global localization problem in large and sparse environments. This problem is difficult since large parts of this kind of environment do not provide relevant structural information for the robot to recover its pose. We propose an initial distribution which puts particles only in relevant parts of environment and lead the robot along relevant regions using the numeric solution of a BVP. The main differences between our proposal compared to the Coastal Navigation [22] is that in our case the robot's initial pose is unknown and our strategy takes advantage of the environment structure as a cue to particle generation.

This paper is organized as follows. Section 2 and 3 present the Monte Carlo Localization and the BVP-path planner, respectively. Section 4 introduces the method proposed to draw particles using BVP and in Section 5 we present an experimental evaluation of the new approach. Finally, Section 6 presents our conclusions.

2. MONTE CARLO LOCALIZATION

Monte Carlo Localization is a class of algorithms that computes a robot's pose using a Bayesian approach and the assumption that the environment is Markovian. Basically, the posterior densities (*beliefs*), $Bel(s_t)$, over the robot state s_t at instant t , are computed, using the sensor data and odometric data in a recursive way through

$$Bel(s_t) = \eta p(o_t|s_t) \int p(s_t|s_{t-1}, a_{t-1}) Bel(s_{t-1}) ds_{t-1} \quad (1)$$

where η is a normalization constant, o_t is the sensor measure gathered during the robot movement at time t and a_{t-1} is the odometric data associated to the robot motion between the instants $t-1$ and t . The $p(s_t|s_{t-1}, a_{t-1})$ is associated to the probabilistic model of the robot motion and $p(o_t|s_t)$ is the probabilistic observation model. Both models assume a Gaussian distribution with zero mean.

To compute Eq. 1 is not a simple task. The main

problem are the computational requirements to calculate $Bel(s)$ over the space of possible poses. To alleviate this problem, Thrun [26] developed a particle filtering that used only a fraction of samples drawn from the poses space.

The filtering represents the beliefs by a set of m samples $S_t = \{s_t^i, w_t^i\}_{i=1}^m$ where each sample represents a particular pose $s^i = (x, y, \theta)$ and the w^i are non-negative importance factors, which sum up to one. The algorithm starts drawing m samples using the initial knowledge about the candidate robot poses given by S_0 . The initial importance factor of each sample is equal to $1/m$, that is, all the samples are equiprobable to be the correct robot pose. Recursively, the new generations of samples are determined according to the steps below

- 1) *Resampling*: Sample s_{t-1}^i from the sample set representing S_{t-1} according to the weights w_{t-1}^i .
- 2) *Sampling*: The odometric data are used to determine the estimated robot pose, s_t^i , at instant t , using $p(s_t|s_{t-1}^i, a_{t-1})$,
- 3) *Importance sampling*: The sample importance w^i is evaluated using

$$w^i = \eta p(o_t|s_t^i)$$

3. PATH PLANNER BASED ON BOUNDARY VALUE PROBLEMS

Recently, we proposed a framework for controlling a robot in navigational tasks, like the exploration of unknown environments and the planning of paths between known positions [6, 7, 28, 21, 8]. It is based on potential fields that do not have local minima generated through the numeric solution of the BVP using Dirichlet boundary conditions and the following equation

$$\nabla^2 p(\mathbf{r}) + \epsilon \mathbf{v} \cdot \nabla p(\mathbf{r}) = 0 \quad (2)$$

where \mathbf{v} is a bias vector and ϵ is a scalar value. The allowed values of the parameters ϵ and \mathbf{v} generate an expressive amount of action sequences that the robot can take to reach a specific target (goal position) or to explore an unknown environment. Basically, the method discretizes the environment into a fixed homogeneous mesh with identical cells, like an occupancy grid. Each cell (i, j) is associated to a squared region of the real environment and possesses a potential value $p_{i,j}$. Dirichlet boundary conditions are such that the cells with high probability of having an obstacle are set to potential 1 (*high*) while cells containing the target are set to potential 0 (*low*).

In the exploratory tasks, the robot collects information on the obstacles using its sensor and updates the mesh

cells. We consider the unknown regions as targets that the robot must reach then the unknown regions cells are assigned to a low potential value (see [21, 28] for more information). By using this information, the Gauss-Seidel algorithm is employed to update the potential cells according to

$$p_{i,j} = \frac{1}{4}(p_{i-1,j} + p_{i+1,j} + p_{i,j-1} + p_{i,j+1}) + \frac{\epsilon}{8}((p_{i+1,j} - p_{i-1,j})v_x + (p_{i,j+1} - p_{i,j-1})v_y) \quad (3)$$

where $\mathbf{v} = (v_x, v_y)$ and $\epsilon \in [-2, +2]$. The robot uses the gradient descent of this potential to determine the path to follow towards the target position. This method is formally complete, i.e., if there is a path connecting the robot position to the target, it will be found. The high potential value prevents the robot from running into obstacles whereas the low potential value generates an attraction basin that pulls the robot.

As the environment that we are handling is sparse, we need strategies that efficiently guide the robot towards the meaningful environment regions. In [28], we extend our strategy to incorporate environmental features. They can be used as good clues to orient the exploration of an unknown environment and to construct routes between locations of interest [24]. By selecting different features in the boundaries we can generate different strategies for the complete exploration of the environment. These strategies are associated with the potential values attributed to the different boundary configurations used to calculate the potential in Eq. 2. For example, consider the environment represented in Figure 1. Regions labeled as “e” are edges of walls. When the robot explores these regions first it should exhibit a wall-following behavior. But if the robot avoids the walls seeking to fill unexplored concave regions, like the ones labeled as “c”, it should exhibit a space-filling behavior.

As we stated above, the behavior of the robot is given by the boundary conditions, where the boundary are held at a potential defined by some function along the boundary, i.e.,

$$p(\mathbf{r}) = f(\mathbf{r}) \quad \text{for } \mathbf{r} \in \partial\Gamma \quad (4)$$

where $\partial\Gamma$ is the borderline between the explored and unexplored space of the environment Γ . Wall-following exploring behavior is achieved just setting a low potential ($f(\mathbf{r}) = 0$) in wall edges and their neighborhood. On the other hand, space-filling exploring behavior is achieved setting low potential in unexplored concave regions.

4. PARTICLE SELECTION BASED ON BVP

A standard particle filter draws samples uniformly at random from the environment free-space. This process

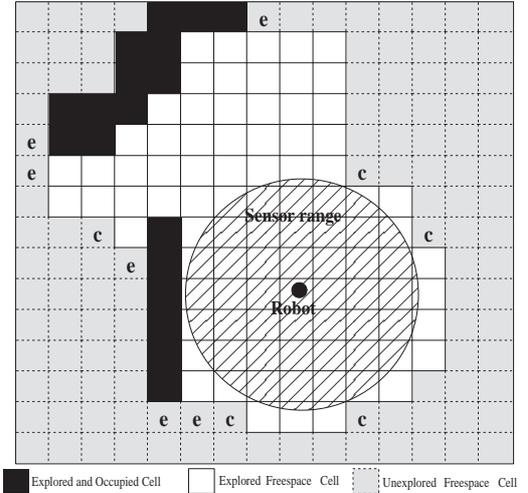


Figure 1. Region of the environment where the BVP is calculated. Searching for edges of walls, regions labeled as e, represents a wall-following exploring behavior and for unexplored concave regions, labeled as c, it corresponds to a space-filling exploring behavior.

has some disadvantages for localization. First, the process can demand a high number of particles to completely cover the environment in order to guarantee that the robot will be able to recover its pose. It is known that the performance of the Monte Carlo filter highly depends on having some particles with a pose close to the real robot pose in the initial distribution [10]. Second, all regions of the environment have the same *a priori* probability to contain the robot pose. In large open regions, the absence of structural information makes the system spend a lot of time processing *irrelevant* particles, that is, particles which cannot contribute to determine the robot pose, since the robot will only be able to localize itself when its sensors capture some information from obstacles. Third, the regions near the obstacles are in general improperly covered with a low particle density that can not represent adequately the real robot pose. Taking these problems into consideration, a better solution is to distribute the particles near obstacles instead of distributing them uniformly over all the environment. To make such a strategy effective, it requires a robust algorithm that leads the robot preferentially along the obstacles while guaranteeing the exploration of the complete environment.

An approach to solve these problems has been proposed by Kwon et al. [16]. They suggest to distribute the particles along the edges of the environment’s topological map. As these edges represent only a fraction of the environment, the method will process a smaller set of particles, thus reducing the computational cost. To localize itself, the robot initially explores the local environment until reaching an edge of the topological map, and after-

wards navigates on the edges of the topological map [15]. The particles' belief is updated only during this second phase.

Kwon et. al.'s method is very adequate when the robot is inserted in a dense environment. As the edges correspond to the middle of the corridors, the robot will navigate the environment always sensing the obstacles around. However in a sparse environment, the edges will conduct the robot through an area where it will not be able to sense any obstacle. Particles on topological edges contribute no information in sparse environments due to the absence of structural information. Then, the method will fail because this area does not have any relevant information that can help the robot to recover its pose.

Our approach solves these problems using a combination of an exploration based on the solution of a BVP [21, 9, 28] and a variation of the skeletization algorithm used in image processing [13]. For localization, the robot first explores the environment until sensing an obstacle. We then use the intermediate results of the skeletization algorithm on the global map to determine relevant parts of the environment which represent all the positions where the robot can sense any obstacle. Afterwards we use the solution of a BVP defined on the skeleton and the global map to generate an initial distribution of the particles. Then, the robot continues the exploration, preferentially following obstacles, using our exploration method described in section 3.

This approach has several advantages. First, the potential produced by the BVP focuses the particle distribution on regions more probable to contain the robot's real pose avoiding large open regions. Second, the BVP does not constrain the robot movement through the middle of corridors (or topological edges), which is not adequate in a sparse environment. The robot will explore, for instance, the environment following walls as long as possible but switch to explore open regions if necessary. Therefore our method can be applied to any kind of environment.

Our proposal can be viewed as an *integrated exploration strategy* [11, 5, 25, 18], since the robot is guided to regions that contain information to minimize the uncertainty about its pose. Strategies like this tend to produce better results than purely random strategies, since they use additional information about the environment to determine the best next action to be executed. They differ mainly in the way they determine the region that must be visited, which can be chosen using a cost function. Amigoni [1] shows that strategies that use cost functions to reach a balance between exploration and precision in mapping tend to be more efficient than random or greedy exploration strategies. As the environment used in the experiment is sparse, regions near obstacles are more important than the others becoming more attractive to the robot.

Then, our approach will lead the robot to the relevant environment region, instead of the regions that represent a sparse area.

The following sections describe our method in detail. Section 4.1 shows how we use skeletization and BVP to find a better initial distribution of the particles. In Section 4.2 we show how we can use the knowledge about the exploration behavior of the robot to further improve the distribution of the particle's initial direction.

4.1. EXTRACTING THE INTERMEDIARY ENVIRONMENT SKELETON

Our method determines the cells that the robot will probably visit during its navigation using the intermediary result of a skeletization process. Skeletization is a morphological thinning used in binary images that successively erodes away pixels from the boundary until no more thinning is possible. At the end, the pixels left over correspond to the image skeleton. In our case, the erosion acts on the free-space cells and at each erosion step, also called relaxation step, the free-space is shrunk at most one cell from its boundary.

If we want to guarantee that the robot always receives sensor readings, we must constrain it to navigate always at a maximum distance d_{robot} from the obstacles. Depending on the environment, the robot can navigate always at a distance d_{robot} . However, the robot may need to visit a narrow corridor with width less than d_{robot} . Then the cells that should be visited comprise the intermediary result of the skeletization process after ρ relaxation steps, where

$$\rho = d_{robot}/w_{cell}$$

and w_{cell} is the grid cell width. Even if the robot enters in a corridor with width less than $2d_{robot}$, the visited cells will be identified by this method. These cells, called *visiting cells*, are marked and receive a potential value of 0.5. The value 0.5 has been chosen to indicate good navigation positions and because the potential values vary in interval [0,1]. Obstacles have a potential value of 1 and cells near the sparse regions have a potential of 0. The skeletization process is continued for another ρ steps and the cells, called *border cells*, that represent the new intermediary skeleton have their potential values set to 0. Then, the harmonic potential of the other cells in the free-space is computed.

Figure 2(a) shows a sparse environment, the robot (solid black circle) and its view field (dashed circle). We can see that the robot can observe only a small fraction of the environment at a time. Figure 2(b) shows the potential field generated using our approach. The cells in red and in blue belong to the intermediary skeleton after the first and second relaxations, respectively. It also shows in gray scale the cells potential value, where the black and white

corresponds to the potential 1 and 0, respectively. The cells in green have a potential value of 0 and a distance of $2d_{robot}$ or more from the obstacles. In this case, the radius of robot vision field is $2d_{robot}$. Hence, the cells in green represent positions where the robot will not sense any obstacle. Figure 2(c) shows the results if we double d_{robot} in comparison to the experiment shown in Figure 2(b). Figure 2(d) shows in red the skeleton extracted after the complete convergence of the skeletization algorithm.

To generate the initial particle distribution, we use the potential value of each cell to determine its probability to be selected to receive particles. Consider that we want to put k particles at a cell $r = (i, j)$. The equation

$$\Pr(r \text{ is selected}) = \exp(-c(p(r) - 0.5)^2) \quad (5)$$

provides the probability that each one of the k particles will be generated at space region related to the cell r , where $p(r)$ is the potential of the cell r and c is a constant that determines what cells around the skeleton will be used to draw the particles based on their potential values. The smaller the constant value the more cells near the skeleton will have high probability to receive particles.

The advantage of using the solution of a BVP to determine the cells' potential is that this method adapts itself to any kind of environment and produces a linear distribution of the potential from the obstacles.

4.2. USING THE POTENTIAL TO IMPROVE FILTER CONVERGENCE

The BVP method allows us to further improve the initial distribution of the particles. Consider the initial phase of the exploration and localization. When the robot starts in a sparse region of the environment, where it cannot sense any obstacle, the BVP path planner will generate a space-filling behavior to explore this open region, until an obstacle is found. Then, the robot will determine its orientation and shows a wall-following behavior, moving perpendicular to the gradient of the current potential field. When encountering the first obstacle, the initial distribution of the particles is generated according to the method described in the previous section. We can improve the likelihood of producing good initial positions taking into consideration the robot's orientation. Since it will follow walls in this phase, particles with orientation perpendicular to the gradient of the global vector field are more likely to represent the robot's actual orientation. We therefore distribute the particle orientations uniformly over all directions which do not deviate more than a maximum angle ϕ from a direction perpendicular to the vector field. The angle ϕ allows to compensate errors due to measurement and incomplete information of the robot's local map. In our experiments, $\phi = 30^\circ$ has shown to produce good results (see section 5.4 below). Figure 3 shows an example of the particle distribution according to this method.

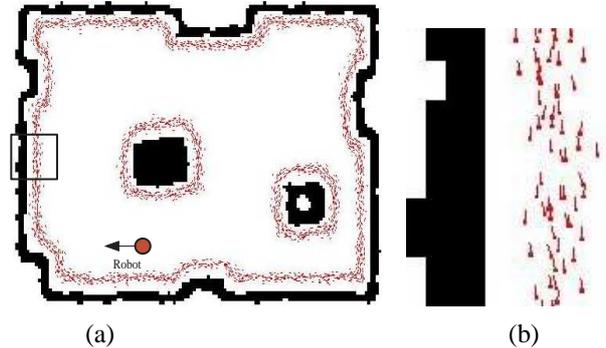


Figure 3. Initial distribution of particles according to initial direction of the robot. (a) Global distribution of the particles and the robot's pose after determining its initial orientation. (b) Detailed view of the particles' distribution in the region highlighted in (a).

5. EXPERIMENTS

This section presents several experiments conducted using the simulator of the robot Pioneer 3DX. In Subsection 5.1, we give examples of the typical behaviour of the global localization process in environments of different density and connectivity. Afterwards, we present the results of three experiments. In a first, preliminary experiment, we compare the ability of the uniform distribution, Kwon et al.'s method and our method based on potential fields to distribute the particles in regions with high relevance for the localization process. The second experiment compares the quality of the localization of the three initial particle distributions. The third experiment studies the improvement of convergence using the method of Section 4.2 to distribute the initial orientation of the particles, according to the robot's orientation. In all experiments, d_{robot} is equal to half of the radius of the robot vision field.

5.1. GLOBAL LOCALIZATION IN ENVIRONMENTS OF DIFFERENT DENSITY

In this subsection, we give three examples of a global localization process using our method based on BVPs for the exploration combined with the improved initial particle distribution described in Section 4. We compare the behavior of the method in environments of different density, as shown in Figures 4(a), 5(a), and 6(a). Environment *A* is dense, environment *B* sparse, but with a connected boundary and environment *C* sparse and non-connected. The environments have been chosen to verify that our method can be applied to any kind of environment, including sparse and non-connected ones, where a simple wall-following strategy will fail to explore the complete environment. The three environments have different sizes and grid cell widths as shown in Table 1. In all three environments, we try to localize the robot, where it can sense any obstacle at a maximum distance of 2 m. The initial

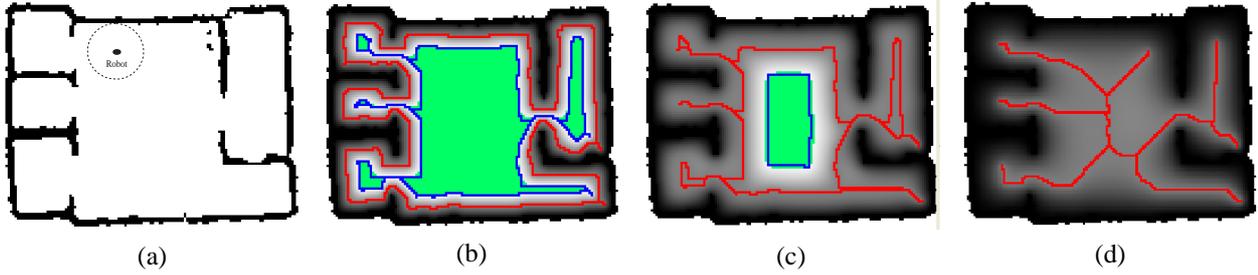


Figure 2. Results of the BVP and skeletonization process in a sparse environment. (a) shows the sparse environment. (b) shows the potential field generated using our approach. The cells in red and in blue belong to the intermediary skeletons. The cells in green are not used to draw the particles. (c) shows the results obtained with a field of vision of double radius d_{robot} compared to (b). (d) shows in red the skeleton extracted after the complete convergence of the skeletonization algorithm.

Table 1. Characteristics of the three environments used in the experiments.

Type	Size [m]	Cell size [cm]	ρ
A Dense	10×10	10×10	10
B Sparse, connected	30×24	24×24	4
C Sparse, unconnected	26×20	24×24	4

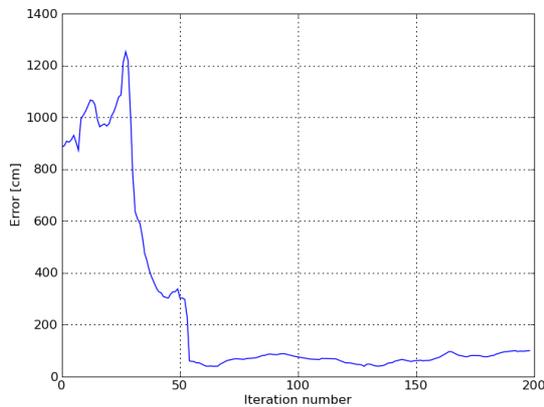


Figure 7. Example of an evolution of the localization error in environment C.

potential field and the resulting initial particle distribution are shown in Figures 4(b), 5(b), 6(b) colored in gray-scale and cyan, respectively.

For environment A, Figures 4(c)-(g) show the particle distribution when the robot is at positions 1-5, respectively, as shown in Figure 4(h). The path followed by the robot during the process is the red curve in Figure 4(h). For environment B, Figures 5(c)-(h) illustrate the convergence of the particles to a limited region in the environment. The path followed by the robot during the process is the red curve in Figure 5(i). For environment C, Fig-

ures 6(c)-(h) show the distribution of the particles when the robot is at positions 1-6, respectively, as shown in Figure 6(i). The path followed by the robot during the process is the red curve in Figure 6(i).

Figure 7 shows a typical example of the evolution of the localization error during the localization process in environment C. The localization error is calculated as the difference between the predicted position of the robot and the actual position in the simulator. The error tends to decrease according to the data acquired by the robot and stay bounded near to zero. In this case, the average distance between the particle has been bounded by 96 cm which corresponds to four times of the environment's cell size.

The magnitude of the localization error in our experiments stems from two main sources:

- The robot maintains a discrete map of the environment, with a varying cell size (see Table 1). Therefore the expected sensor reading of a particle can be determined to a precision of at best one cell size.
- To determine the true localization error in the simulation, we have to map the current position estimate of the robot's position, into the simulator's coordinate system. This is done using a linear transformation. It can introduce an additional, artificial error, if the robot's map contains non-linearities introduced during the initial exploration.

For this reason, in the following experiments, we consider a localization trial being successful, if the final localization error is less than four times the map's cell size. Observe that the error introduced by the discretization can be decreased by reducing the cell size during exploration, at a higher processing cost.

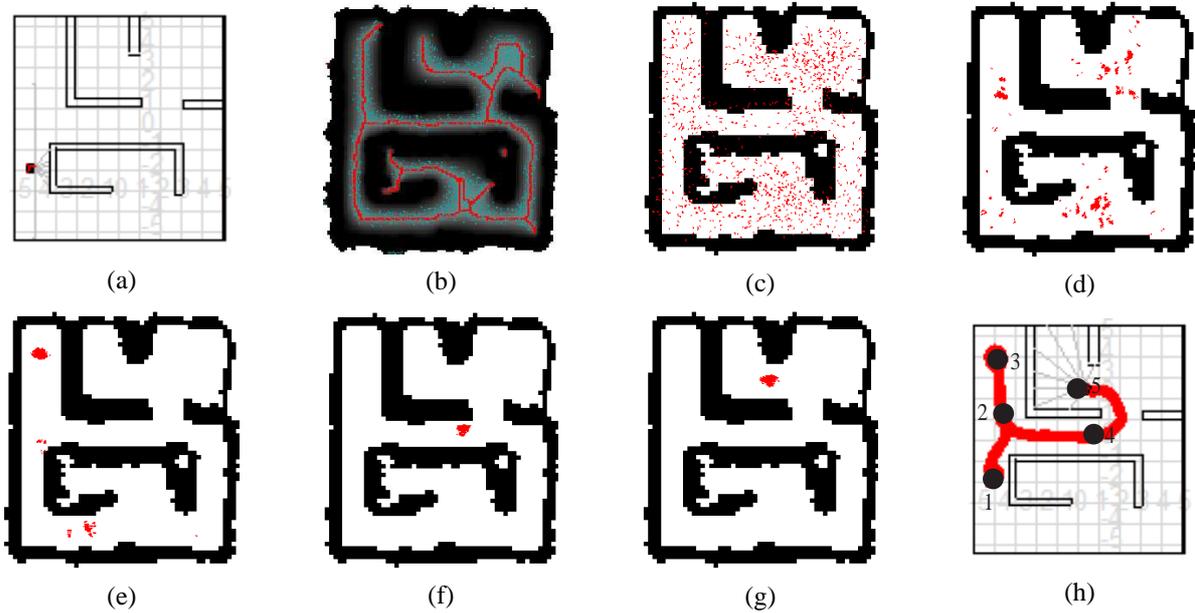


Figure 4. Example of a successful localization in environment A. (a) Map of the environment. (b) Potential field (in grey) used to define the initial particle distribution (in cyan). (c)-(g) Particle distribution (in red) when the robot is at positions 1-5 and simulation steps 1, 11, 21, 60, 76, respectively, as shown in (h).

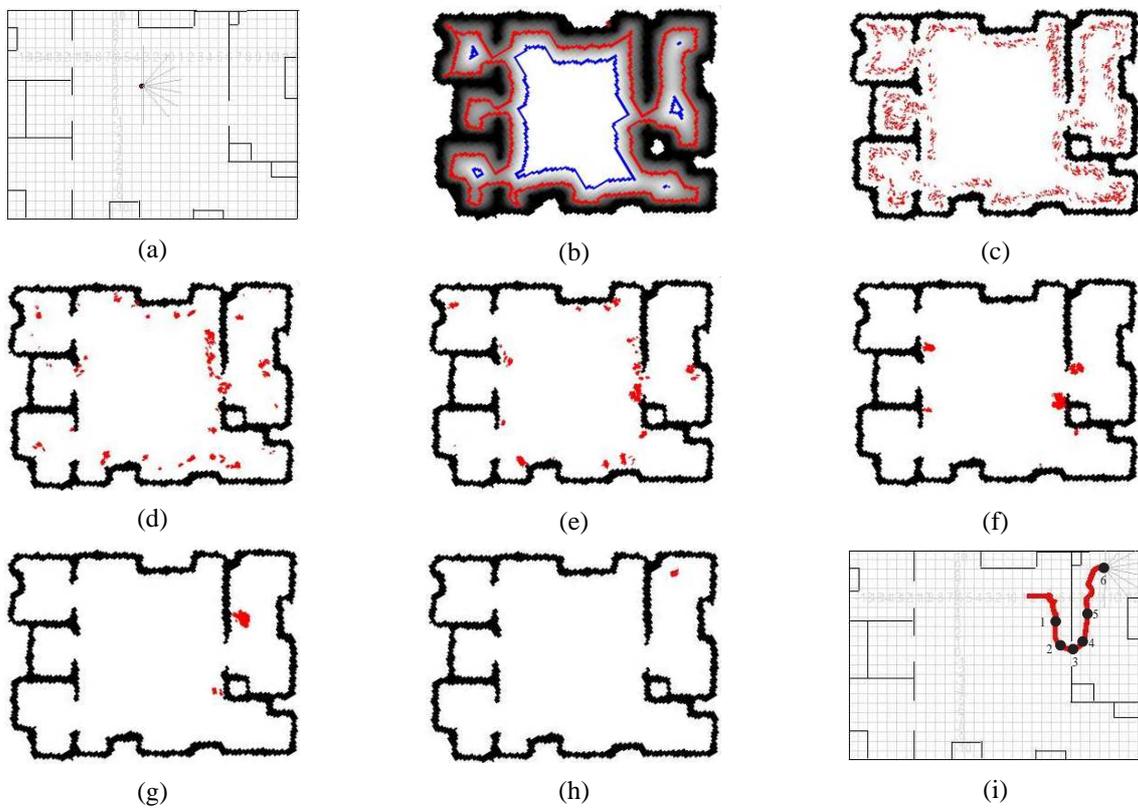


Figure 5. Example of a successful localization in environment B. (a) Map of the environment. (b) Potential field (in grey) used to define the initial particle distribution (in cyan). (c)-(h) Particles distribution (in red) when the robot is at positions 1-6 and simulation steps 1, 14, 20, 26, 39, 65, respectively, as shown in (i).

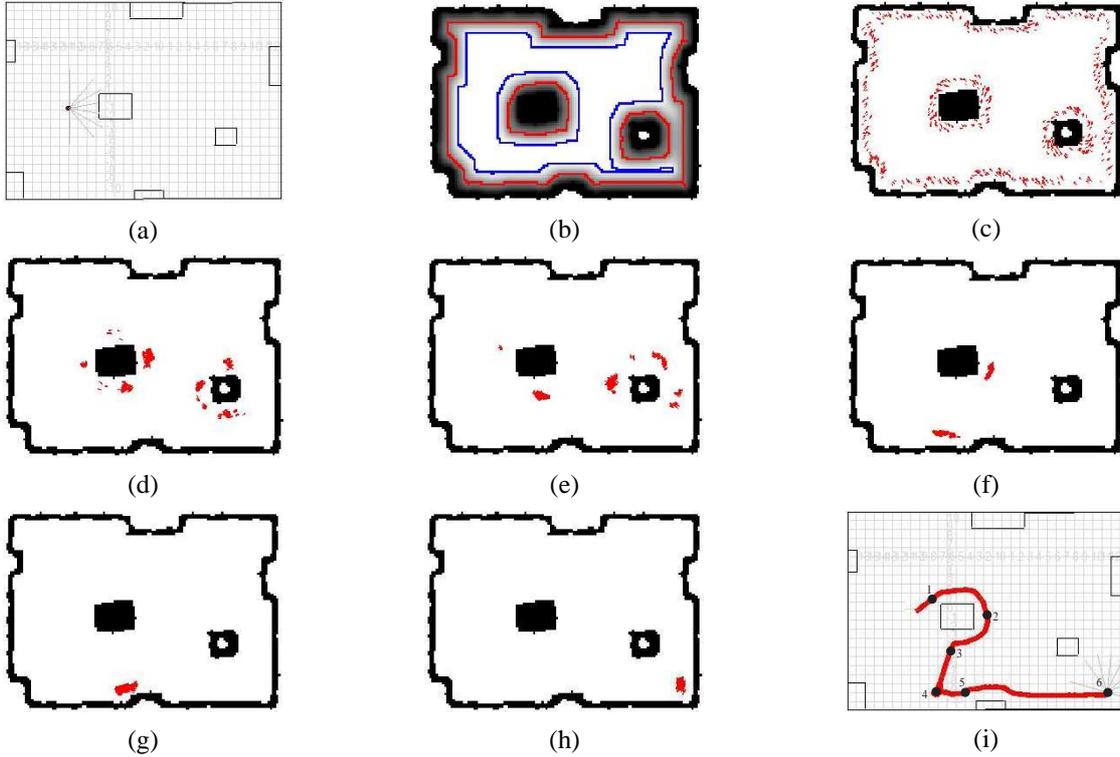


Figure 6. Example of a successful localization in environment C. (a) Map of the environment. (b) Potential field (in grey) used to define the initial particle distribution (in cyan). (c)-(h) Particle distribution (in red) when the robot is at positions 1-6 and simulation steps 2, 38, 65, 95, 108, 183, respectively, as shown in (i).

5.2. EXPERIMENT 1: COMPARISON OF THE INITIAL PARTICLE DISTRIBUTION OF THE THREE METHODS

With this preliminary experiment we want to verify that our method produces good environment coverage when compared to the uniform distribution and the topological distribution. To this aim, we focus this experiment on the particle distribution generated by each method, more precisely, in the amount of selected cells that will contain the particles instead of the amount of particles per cell. So, we consider that each free-space cell can be selected or not to contain just one particle.

The uniform distribution selects a free-space cell with probability 0.5. The topological distribution selects cells that comprise the environment’s topological map or neighboring cells up to a distance of two cells with probability 0.5. Our method chooses a cell based on its potential value using Eq. 5 and $c = 20$, that is, the cells with potential values in the interval $[0.4, 0.6]$ are more likely to be used to draw the particles.

The experiments have been conducted in the environment shown in Figure 2(a) since it contains sparse and dense regions depending on the robot’s field of vision. If the robot’s field of vision is small, then the environment is completely sparse, otherwise it can be dense inside the

rooms and sparse in the center, or dense everywhere.

To compare the resulting particle distributions, we determine the amount of free-space cells that can provide sensor information, to help the robot to recover its pose. Figure 8 shows an example of the cells selected using the three methods. The cells in green have been selected using the uniform distribution, while the cells in red and in lilac have been selected using our method and the topological distribution, respectively. Considering that the robot has a small field of vision, as shown in Figure 2(a), we can see that most of cells selected by the uniform distribution and the topological distribution will be placed in the center of the environment. On the other hand, our method selects the cells near the robot that are more likely to correspond to the real robot pose.

These observations are confirmed by the following experiment. We subdivide the real environment of approximately $30\text{ m} \times 24\text{ m}$ into grid cells of size $24\text{ cm} \times 24\text{ cm}$. We consider that the robot navigates at maximum distance d_{robot} from the obstacles and that the radius of vision field is $2d_{robot}$. The distance d_{robot} has been varied from 50 cm up to 2.5 m in increments of 25 cm. For each value cells relevant for localization are at maximum distance $2d_{robot}$ from the obstacles, since cells more distant than this comprise the sparse region of the environment. The

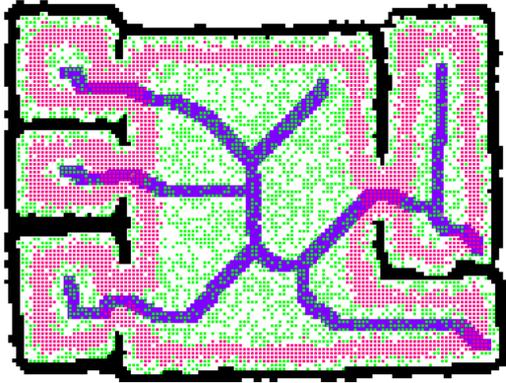


Figure 8. Particle density in the environment using the three distributions. The cells in green have been selected using the uniform distribution. The cells in red and in lilac have been selected using our method and the topological distribution, respectively.

environment has in average over different fields of vision 4520 relevant cells and 5250 irrelevant cells.

A comparison of the three methods is shown in Figure 9. The uniform distribution selected in average 2260 relevant particles and 2650 irrelevant ones. The relevant cells correspond to $\approx 43\%$ of relevant free-space cells. The topological distribution selected in average 571 relevant particles and 821 irrelevant ones. The relevant cells correspond to $\approx 12\%$ of relevant free-space cells. Our method selected $\approx 63\%$ of the relevant free-space cells and only 0.3% of irrelevant cells. The processing using our method focuses just on the regions that can contribute for the robot to recover its pose.

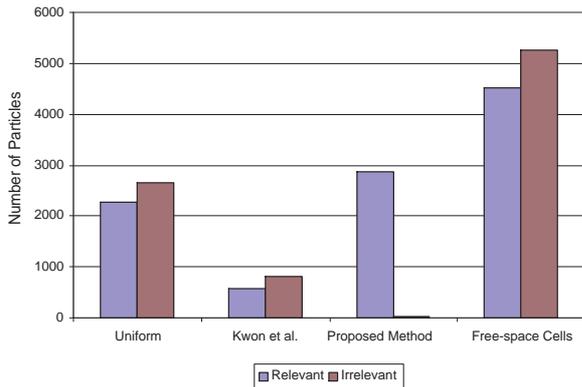


Figure 9. Comparison of relevant and irrelevant particles selected by different methods.

In conclusion, our method is able to generate an initial particle distribution which almost entirely consists of positions relevant for the localization of the robot. This leads to a higher convergence rate, as the following experiments show.

Table 2. Percentage of successful localizations. For each initial particle distribution and number of particles the convergence rate without and with orientation of the particles is given.

Distribution	Number of particles			
	2k	4k	6k	8k
Uniform	18 / 43	22 / 62	45 / 77	50 / 66
Topological	10 / 20	20 / 22	16 / 20	20 / 25
Our method	43 / 58	58 / 75	56 / 83	66 / 79

5.3. EXPERIMENT 2: INFLUENCE OF THE INITIAL DISTRIBUTION ON THE CONVERGENCE

In this set of experiments we compare the convergence of the Monte Carlo filter for different initial particle distributions, where the particle orientation is random. The experiments have been conducted in the environment shown in Fig. 6(a). This environment is challenging for a global localization, since it is sparse, non-connected and has several symmetries.

In each experiment we generated 50 different initial positions of the robot. For each position we try to localize the robot using the uniform distribution, the topological distribution and our method, with sample sizes of 2000, 4000, 6000 and 8000 particles. Each localization trial has been run for 150 time steps. A trial is considered to be successful, when the difference between the predicted and actual robot position is less than 96 cm (four times the cell size) at the end of the run. A trial which did not converge in this time frame is considered a failure. Fig. 10(a) and Table 2 show the convergence rate, i.e. the number of successful localizations, as a function of the number of particles and the initial particle distribution.

The chosen environment is large and sparse, and therefore the probability of generating an initial particle close to the true position of the robot is small. Since, the performance of the Monte Carlo filter highly depends on this (see the discussion in section 4), we observe low overall convergence rates between 10% and 66%. As expected, the convergence rate increases with the number of particles for all three initial distributions. The proposed method performs better than the other distributions, with 11% to 46% more successful localizations. In particular, it obtains with 4k particles a better convergence rates than the uniform distribution with 8k particles.

The average final localization error of the successful localizations is shown in Fig. 10(b). For all three methods, and independent of the number of particles, a successful localization leads to an average localization error of about 63 cm equivalent to 2.6 cells, with a standard deviation of less than one cell width.

For a large number of particles we expect the uniform distribution and our method to converge to an always successful location in all trials. To confirm this, we

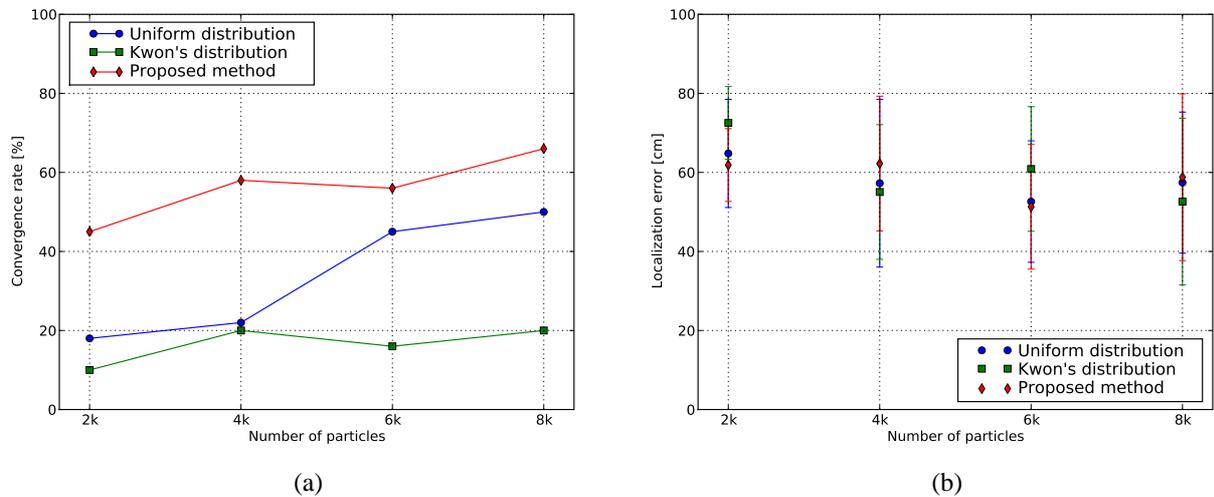


Figure 10. Results of localization experiments for different initial particle distributions and different numbers of particles. (a) Percentage of successful localizations. (b) Localization error of successful localizations.

performed one additional experiment with 20k particles, and obtained convergence rates of 75% and 85% for the uniform distribution and our method, respectively.

5.4. EXPERIMENT 3: INFLUENCE OF THE IMPROVED INITIAL ORIENTATION OF THE PARTICLES

In this set of experiment we study the convergence rate when distributing the orientation according to the known initial orientation of the robot as described in section 4.2, and compare it to the strategy of the previous experiment, which distributes the orientation uniformly over all possible angles. The experimental setup is the same as in the previous experiment: we report averages over 50 trials, for each initial particle distribution, and samples sizes of 2000, 4000, 6000 and 8000 particles. All experiments have been conducted in the environment shown in Figure 6(a) choosing the robot's initial pose randomly in the environment.

From the results shown in Figure 11 and Table 2 we can see that the use of the vector field significantly improves the convergence rate of the particle filter. The main reason is that it adds information on the expected robot pose during the localization process instead of using random particles with low likelihood of being a real robot pose. Generating the particles without regarding the expected robot behavior is to produce useless particles that will tend to rapidly disappear and will not contribute to the localization only increasing the processing cost. For instance, our method, using the vector field allows the filter to converge with 4000 particles in 75% of the trials (Figure 11(a)). Without using the information about the initial orientation, the standard Monte Carlo filter with uniform initial distribution achieves only a con-

vergence rate of 66% with twice the number of particles (Figure 10(a)).

As in the previous experiment, we performed one additional evaluation with 20k particles to verify the convergence of the methods. For both, the uniform and the proposed method, we obtained a convergence rate of 85% in this experiment.

6. CONCLUSIONS

In this paper, we propose a strategy that combines a path planner based on boundary value problems and MCL to solve the global localization problem in sparse environments. The proposed method has the advantage, that the potential produced by the BVP focuses the particle distribution on regions more probable to contain the robot pose avoiding large open regions. Furthermore, the BVP path planner does not constrain the robot's motion through the middle of corridors and can be used in dense as well as sparse environments. The localization driven by the solution of a BVP on the environment guides the robot along its significant parts. Our experiments show, that we can make a better use of the knowledge about the initial position in the environment extracted from the potential than the particle filter alone, improving the convergence rate about 10% to 20% in a sparse environment. The localization is precise up to about four times the resolution of the underlying grid. This allows to achieve the same convergence rates and localization errors using fewer particles.

For future work, we intend to deal with symmetric environments and to use the information collected on the environment to determine where the robot must explore in order to minimize the odometric errors and produce a

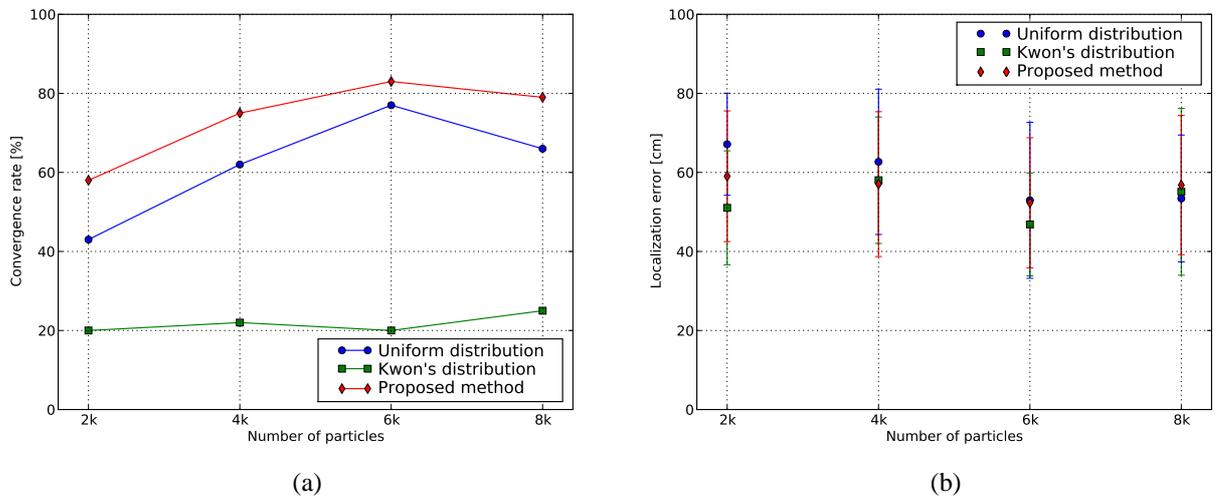


Figure 11. Results of localization experiments for different initial particle distributions and different numbers of particles. The particle's initial orientation is distributed using the orthogonal vector field according to the method proposed in Section 4. (a) Percentage of successful localizations. (b) Localization error of successful localizations.

trustful environment map.

Acknowledgments

This work has been partially supported by Fundação de Amparo à Pesquisa do Rio Grande do Sul (FAPERGS) and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

References

- [1] Francesco Amigoni. Experimental evaluation of some exploration strategies for mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2008.
- [2] Adrien Angeli, Stéphane Doncieux, Jean-Arcady Meyer, and David Filliat. Incremental vision-based topological slam. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008.
- [3] Momotaz Begum, George K. I. Mann, and Raymond G. Gosine. Concurrent mapping and localization for mobile robot using soft computing techniques. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2005.
- [4] Jose-Luis Blanco, Juan-Antonio Fernández-Madrugal, and Javier González. Efficient probabilistic range-only slam. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008.
- [5] Frederic Bourgault, Alexei A. Makarenko, Stefan B. Williams, Ben Grocholsky, and Hugh F. Durrant-Whyte. Information based adaptive robotic exploration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2002.
- [6] F. Dapper, E. Prestes, M. A. P. Idiart, and L. P. Nedel. Simulating pedestrian behavior with potential fields. In *Advances in Computer Graphics*, volume 4035 of *Lecture Notes in Computer Science*, pages 324–335. Springer Verlag, 2006.
- [7] F. Dapper, E. Prestes, and L. P. Nedel. Generating steering behaviors for virtual humanoids using bvp control. In *Advances in Computer Graphics*, Lecture Notes in Computer Science. Springer Verlag, 2007.
- [8] Gedson Faria, Edson Prestes, Marco A. P. Idiart, and Roseli A. F. Romero. Multi robot system based on boundary value problems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 424–429, 2006.
- [9] Gedson Faria, Roseli Aparecida Francelin Romero, Edson Prestes, and Marco A. P. Idiart. Comparing harmonic functions and potential fields in the trajectory control of mobile robots. In *IEEE Conference on Robotics, Automation and Mechatronics*, volume 1, pages 762–767, Singapore, December 2004.
- [10] D. Fox, S. Thrun, F. Dellaert, and W. Burgard. *Sequential Monte Carlo in practice*, chapter Particle filters for mobile robot localization. Springer, 2001.

- [11] Luigi Freda, Francesco Loiudice, and Guisepppe Oriolo. A randomized method for integrated exploration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006.
- [12] Andrea Gasparri, Stefano Panziera, Federica Pascucci, and Giovanni Ulivi. Monte carlo filter in mobile robotics localization: A clustered evolutionary point of view. *Journal of Intelligent and Robotic Systems*, 47(2):155–174, 2006.
- [13] R. Gonzalez and R. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, 1992.
- [14] Patric Jensfelt and Steen Kristensen. Active global localisation for a mobile robot using multiple hypothesis tracking. *IEEE Transactions on Robotics and Automation*, 17(5):748–760, 2001.
- [15] Tae-Bum Kwon and Jae-Bok Song. Thinning-based topological exploration using position probability of topological nodes. In *IEEE International Conference on Robotics and Automation*, 2006.
- [16] Tae-Bum Kwon, Ju-Ho Yang, Jae-Bok Song, and Woojin Chung. Efficiency improvement in monte carlo localization through topological information. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 424–429, 2006.
- [17] J.J. Leonard and H.F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation*. Kluwer Academic, 1992.
- [18] Cindy Leung, Shoudong Huang, and Gamini Dissanayake. Active slam using model predictive control and attractor based exploration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2006.
- [19] Adam Milstein, Javier N. Sanchez, and Evan T. Williamson. Robust global localization using clustered particle filtering. In *Innovative Applications of Artificial Intelligence (IAAI)*, pages 581–586, 2002.
- [20] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *National Conference on Artificial Intelligence (AAAI)*, Edmonton, Canada, 2002. AAAI.
- [21] Edson Prestes e Silva Junior, Paulo M. Engel, Marcelo Trevisan, and Marco A.P. Idiart. Exploration method using harmonic functions. *Robotics and Autonomous Systems*, 40(1):25–42, 2002.
- [22] Nicholas Roy, Wolfram Burgard, Dieter Fox, and Sebastian Thrun. Coastal navigation – mobile robot navigation with uncertainty in dynamic environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, May 1999.
- [23] B. Schiele and J.L. Crowley. A comparison of position estimation techniques using occupancy grids. In *IEEE International Conference on Robotics and Automation*, 1994.
- [24] L. Smith and P. Husbands. Visual landmark navigation through large-scale environments. In *EPSRC/BBSRC International Workshop on Biologically-Inspired Robotics: The Legacy of W. Grey Walter*, pages 272–279, 2002.
- [25] Cyrill Stachniss, Giorgio Grisetti, and Wolfram Burgard. Information gain-based exploration using rao-blackwellized particle filters. In *Proceedings of Robotics: Science and Systems*, Cambridge, 2005.
- [26] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2000.
- [27] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT, 2005.
- [28] Marcelo Trevisan, Marco A. P. Idiart, Edson Prestes, and Paulo M. Engel. Exploratory navigation based on dynamic boundary value problems. *Journal of Intelligent and Robotic Systems*, 45(2):101–114, 2006.