J Intell Robot Syst (2006) 47: 155–174
DOI 10.1007/s10846-006-9081-1

Monte Carlo Filter in Mobile Robotics Localization: A Clustered Evolutionary Point of View

Andrea Gasparri · Stefano Panzieri · Federica Pascucci · Giovanni Ulivi

Received: 3 August 2006 / Accepted: 2 September 2006 /
Published online: 30 September 2006

Abstract Localization, i.e., estimating a robot pose relative to a map of an environment, is one of the most relevant problems in mobile robotics. The research community has devoted a big effort to provide solutions for the localization problem. Several methodologies have been proposed, among them the Kalman filter and Monte Carlo Localization filters. In this paper, the *Clustered Evolutionary Monte Carlo* filter (CE-MCL) is presented. This algorithm, taking advantage of an evolutionary approach along with a clusterization method, is able to overcome classical MCL filter drawbacks. Exhaustive experiments, carried on the robot ATRV-Jr manufactured by iRobot, are shown to prove the effectiveness of the proposed CE-MCL filter.

Key words clustering · genetic algorithms · Monte Carlo integration methods · robot localization

1 Introduction

In mobile robotics, one of the most important goal is to realize the complete autonomy of the robot. The availability of reliable pose information is fundamental to achieve such autonomy.

A. Gasparri · S. Panzieri · F. Pascucci (✉) · G. Ulivi
Dipartimento di Informatica e Automazione, Universitá "Roma Tre",
Via della Vasca Navale 79, 00146 Roma, Italy
e-mail: pascucci@dia.uniroma3.it

A. Gasparri
e-mail: gasparri@dia.uniroma3.it

S. Panzieri
e-mail: panzieri@dia.uniroma3.it

G. Ulivi
e-mail: ulivi@dia.uniroma3.it

The localization problem aims to estimate the robot's pose in an environment, using data coming from sensors. The interaction between the robot and the environment, along with the presence of noisy sensors readings make the problem even more difficult.

The majority of works in literature relies on the probabilistic framework. The idea underlining such approaches is to recursively maintain a probability distribution, called *Belief*, over all poses (state space points) in the environment.

Initially, the research community has been oriented toward the position tracking problem, successively other problems, such as the global localization and the kidnap problem have been investigated as well.

Position tracking is the problem of estimating the robot's pose with prior knowledge about the initial robot's location. The Kalman filter approach is a widespread probabilistic framework successfully employed to face this problem [1].

Kalman based methods represent the *Belief* by means of a Gaussian distribution over the state space of the robot. The mode of the distribution yields the current robot position, while the variance represents the accuracy of the estimation.

Gaussian distribution, described by means of only two parameters, has two important advantages: from a mathematical point of view a discretization of the state space is no longer required [2], while from a computation standpoint, an on line implementation can be easily faced [3].

On the other hand, the gaussian assumption limits the Kalman filter usability when the ability to represent multi-hypotheses is required, as in the *Global Localization* problem.

Global Localization is the problem of estimating the robot's pose without benefit of prior knowledge of the initial robot's location. This lack of knowledge makes the problem even more difficult as environmental ambiguities have to be carefully considered in order to successfully determine the initial robot's pose from scratch.

To overcome such limitations, several probabilistic global techniques have been proposed in literature, relaxing Gaussian assumption and introducing different methods for the discretization of the state space.

In [4] a grid based discretization of the state space has been proposed to localize the robot. The underlining idea is to build a prior global occupancy grid to compare with a local one, built by the robot as time passes. The advantage of the grid-based approach is that it approximates a more complex distribution. However, it suffers from excessive computational overhead [5, 6].

A more promising approach is based on sequential Monte Carlo integration methods [7]. These methods were first investigated in the early 70s [8–10]. However, because of the lack of computational resources available in that period, these techniques were neglected till the 90s when, thanks to a substantial technological improvement, these methods have been rediscovered.

Nowadays, Monte Carlo techniques are successfully applied to solve estimation problems in several research areas, such as computer vision [11], wireless telecommunications [12] and mobile robot localization [13].

The key idea is to use a set of random weighted samples to approximate the probability distribution. The advantage consists in the possibility to represent a large number of probability distributions. Furthermore, the higher the number of particles used, the better the approximation obtained.

However, Monte Carlo integration methods suffer from *degeneracy problem*, i.e., the problem of having most of the particles with a negligible weight after few iterations [14].

This problem turns out to be critical when using Monte Carlo integration methods to deal with the *kidnap problem*, i.e., the problem of having new data that force the estimation of an already localized robot at a completely different position.

In order to avoid the degeneracy, several approaches have been proposed in the literature. An easy way to reduce this phenomenon is to use a large number of particles, however this approach cannot be applied due to the computational effort.

A more effective solution is to introduce a different measure of degeneracy in order to perform a resampling step whenever a significant degeneracy is observed [15].

Moreover, to further reduce this phenomenon, a suitable candidate of the importance function has been suggested in [16]. In particular, the best candidate is the one who minimizes the variance of the importance weights conditioned upon the data.

Two interesting algorithms have been also proposed in [17] and [18]. These works extend the classical Monte Carlo integration methods introducing alternative techniques to obtain particular properties.

The former introduces the idea of clusters of particles to track multiple distinct hypotheses, where each cluster is considered as an independent hypothesis about the robot's pose. The algorithm works on two different levels: at particle level, the classical Bayesian formulation is adopted to update an hypothesis, while at cluster level, the one with the highest probability is used to determine the robot's pose. Despite the ability to maintain distinct hypotheses over time, having a constant number of clusters limits the possibility to solve the kidnap problem because, as the robot moves, the coverage of the environment is no longer guaranteed.

The latter introduces an alternative re-sampling schema, based on genetic algorithms, to mitigate the sample impoverishment problem. The algorithm is able to maintain the diversity of particles during the resampling process by means of the crossover genetic operator. However, it has been conceived only to solve the position tracking problem on a landmark-based framework.

In this paper, an alternative Monte Carlo Filter is proposed. This algorithm, taking advantage of an evolutionary approach along with a clusterization method is able to solve both the global localization problem and maintain the multi-hypotheses. The idea is to exploit a dyanamical clustering for a better data-driven exploration of the search space, and apply a local evolutionary approach, within each subset, for a quicker location of the best solutions. In fact, performing a smart partition of the search space leads to a more efficient use of the evolutionary approach. In detail, from a local point of view, performing a research of the best hypotheses in a subset, results in a faster location of the maxima as well as in a reduction of the probability to stall in a local solution. Again, from a global standpoint, having several clusters in which the evolutionary approach can be run, leads to an implicit parallelization of the exploration. Moreover, the dynamical nature of clusters, can guarantee a better coverage of the environment, allowing at each iteration to focus the attention only where the probability to find the real robot is higher. A preliminary analysis of this approach has been accomplished in [19].

The paper is organized as follows. In Section 2 an introduction to the basic concepts exploited in this work is provided. In Section 3 the proposed Clustered Evolutionary Monte Carlo filter (CE-MCL) filter is explained. In Section 4 the experimental results are shown and discussed. Finally, in Section 5 conclusions are presented and future works are proposed.

2 Theoretical Background

2.1 Probabilistic Framework

A suitable framework for the localization problem can be devised exploiting the probability theory. From a probabilistic point of view, the robot's pose can be described by a probability distribution called *Belief*. As a result, the localization problem consists of estimating the *Belief* over the state space conditioned on the data.

A Bayesian framework to estimate this probability distribution, called *Markovian Framework*, has been introduced in [20]. The key idea is to recursively compute the *Belief* by means of the Bayes rule as new sensors measurement comes.

In literature the *Belief* is defined as:

$$Bel(x_k) = p(x_k \mid D_k),\ x \in \Xi, \tag{1}$$

It represents the probability to have the robot at location x_k at time k, given all the data D_k up the time, where Ξ is the set of all poses.

In mobile robotics, data can be broken down into *control data* and *perceptual data*. Control data represents the inputs of the system and, as they are not always known, are retrieved by encoders or other proprioceptive sensors. Perceptual data represents information about the environment, such as laser measurements.

As a consequence, *prior* and a *posterior belief* can be defined as follow:

$$Bel^{-}(x_k) = p(x_k \mid U_{k-1}, Z_{k-1}). \tag{2}$$

that it is the belief the robot has got, after the integration of the control data u_{k-1}, and before it receives the perceptual data z_k.

$$Bel^{+}(x_k) = p(x_k \mid U_{k-1}, Z_k). \tag{3}$$

that is the belief the robot has got once the perceptual data z_k has been integrated.

Regarding to the integration data, several considerations need to be made:

1) Using the *Total Probability Theorem* the $Bel^{-}(x_k)$ can be rewritten as:

$$\begin{aligned} Bel^{-}(x_k) = \int_{\Xi} & p(x_k \mid x_{k-1}, U_{k-1}, Z_{k-1}) \\ & \times p(x_{k-1} \mid U_{k-1}, Z_{k-1}) dx_{k-1}. \end{aligned} \tag{4}$$

   The equation states that the prior belief of being in state x_k is the sum of the probabilities of coming from state x_{k-1} to state x_k given all the earlier sensor measurements. The second term of the integral represents the *Belief* at time $(k-1)$, as the robot pose at generic step k does not depend on the action that is performed at the same step. To further simplify the formulation, the assumption to have a *Markov* environment can be introduced. The key idea is to considerthe past and future data independent, with the knowledge of the current state [20]. As a consequence the *prior* can be written as:

$$Bel^{-}(x_k) = \int_{\Xi} p(x_k \mid x_{k-1}, u_{k-1}) \times Bel^{+}(x_{k-1}) dx_{k-1}. \qquad (5)$$

2) Using the *Bayes rule* the *posterior* can be rewritten as:

$$Bel^{+}(x_k) = p(z_k \mid x_k, U_{k-1}, Z_{k-1}) \times \frac{p(x_k \mid U_{k-1}, Z_{k-1})}{p(z_k \mid U_{k-1}, Z_{k-1})} \qquad (6)$$

The equation states that the posterior belief is the conditional probability of observing z_k, weighted by the ratio of the prior belief of being in state x_k, $Bel^{-}(x_k)$, and the probability of observing measurement z_k conditioned on all information so far. To further simplify the formulation, the *Markov* assumption can be adopted again. As a result the *Posterior* can be rewritten as:

$$Bel^{+}(x_k) = \frac{p(z_k \mid x_k)\ Bel^{-}(x_k)}{p(z_k \mid U_{k-1}, Z_{k-1})} \qquad (7)$$

3) As a combination of the equations mentioned above, the recursive formulation for localization is:

$$Bel^{+}(x_k) = \eta p(z_k \mid x_k) \times \int_{\Xi} p(x_k \mid x_{k-1}, u_{k-1}) Bel^{+}(x_{k-1}) dx_{k-1}, \qquad (8)$$

where η represents $P(z_k \mid U_{k-1}, Z_{k-1})$ and can be viewed as a normalization factor.

As the integrals above are not tractable, several efforts have been devoted to approximate the state space in order to make the recursive equation above simple to be computed.

2.2 Monte Carlo Integrations Methods

Monte Carlo integrations methods extend the Markovian framework by means of a sampling approach to represent the posterior distribution (Belief). These methods have the significant advantage of not being subject to any linearity or Gaussianity constraints on the model, and they also offer interesting convergence properties. As a consequence, these methods turn out to be a powerful tool to deal with the global localization problem.

The *Perfect Monte Carlo Sampling* draws N independent and identically distributed random samples $\{x_k^{(1)}, \ldots, x_k^{(N)}\}$ according to $Bel^{+}(x_k)$. Consequently, the approximation of the posterior is given by

$$Bel^{+}(x_k) \approx \frac{1}{N} \sum_{i=1}^{N} \delta_{x_k^{(i)}} \left(x_k - x_k^{(i)} \right), \qquad (9)$$

where $\delta_{x_k^{(i)}}\left(x_k - x_k^{(i)}\right)$ represents the delta-Dirac mass located in $x_k^{(i)}$.

However, due to the difficulty of efficiently sampling from the posterior distribution $Bel^+(x_k)$ at any sample-time k, a different approach is required.

An alternative solution is the *Sequential Importance Sampling* approach. The key idea is of drawing samples from a normalized *importance sampling distribution* $\pi(x_k \mid d_k)$ which has a support including that of the posterior $Bel^+(x_k)$. In this case, the approximation of the posterior is given by

$$Bel^+(x_k) \approx \sum_{i=1}^{N} w_k^{(i)} \delta_{x_k^{(i)}}\left(x_k - x_k^{(i)}\right), \tag{10}$$

where the *importance weight* is computed as

$$w_k^{(i)} = w_{k-1}^{(i)} \cdot \frac{P\left(z_k \mid x_k^{(i)}\right) Bel^-(x_k)}{\pi(x_k \mid d_k)}. \tag{11}$$

In mobile robotics, a suitable choice of the importance sampling distribution $\pi(x_k \mid d_k)$ is the prior distribution $Bel^-(x_k)$ [21]. With this choice, the importance weight can be easily computed as:

$$w_k^{(i)} = w_{k-1}^{(i)} \cdot P\left(z_k \mid x_k^{(i)}\right), \tag{12}$$

and the importance sampling distribution can be written in a recursive fashion:

$$\pi(x_k \mid d_{k-1}) = P(x_k \mid x_{k-1}, u_{k-1}) \cdot Bel^+(x_{k-1}). \tag{13}$$

Such formulation has the advantage of allowing an on-line evaluation of the importance weight as long as new data is available; however it causes the degeneracy problem, i.e., the problem of having most of the samples with a negligible weight after few iterations. A common approach to overcome this problem is to provide a resampling step, which aims to replace particles with small importance weight by means of a suitable strategy.

The Algorithm 1 shows a typical implementation schema for a *Sequential Monte Carlo filter with resampling step*. The majority of works in literature relies on this schema, with a specialization for the resampling approach adopted.

2.3 Genetic Algorithms

Genetic Algorithms (GAs) are a class of search methods and computational models inspired by Darwin's *Theory of Evolution*. These algorithms, initially investigated in [22], use a population to explore the search space, by means of probabilistic transition operators like crossover and mutation, in order to find out the element (*chromosome*) that best fits a given objective function (*fitness function*). This approach reflects a possible mathematical model of the *nature's behavior* in which the high adaptability of each creature in its environment is the result of a long evolutionary process, based on natural selection, mutation, sexual and asexual reproduction [23].

GAs have been applied in several research areas to solve optimization problems where the presence of non-differentiable or non-continuous objective functions makes other methodologies almost useless.

Algorithm 1: Sequential Monte Carlo Filter

```
Data: $Bel^{+}(x_{k-1}) = \{x^{i}_{k-1}, w^{i}_{k-1}\}, u_{k-1}, z_k$
Result: $Bel^{-}(x_k)$

/* Importance Sampling                                              */
Compute $\pi(x_k \mid d_{k-1}) = p(x_k \mid x_{k-1}, u_{k-1}) \cdot Bel^{+}(x_{k-1})$
for $i = 1$ to $Ns$ do
    Sample $\tilde{x}^{(i)}_k \sim \pi(x_k \mid d_{k-1})$
    Evaluate $w^{*(i)}_k = w^{*(i)}_{k-1} \cdot \frac{p(z_k|\tilde{x}^{(i)}_k) Bel^{-}(x_k)}{\pi(x_k|d_{k-1})}$
end
/* Normalization                                                    */
for $i=1$ to $Ns$ do $\tilde{w}^{(i)}_k = \frac{w^{*(i)}_k}{\sum_{j=1}^{Ns} w^{*(i)}_k}$
Evaluate $N_{eff} = \frac{1}{\sum_{i=1}^{Ns} (\tilde{w}^{(i)}_k)^2}$

/* Degeneracy Test                                                  */
if $N_{eff} \geq N_{thres}$ then
    $\{x^{(i)}_k, w^{(i)}_k\} = \{\tilde{x}^{(i)}_k, \tilde{w}^{(i)}_k\}$
else
    /* Resampling                                                   */
    $\{x^{(i)}_k, w^{(i)}_k\} = ResamplingProcedure(\{\tilde{x}^{(i)}_k\ ,\ \tilde{w}^{(i)}_k\})$
end
$Bel^{+}(x_k) = \{x^{(i)}_k, w^{(i)}_k\}$
```

A simple genetic algorithm, as it is referred to in [24], can be described as a sequence of the following steps:

- Initialization
- Generation.

Initialization generates an initial population whose elements are encoded by means of a fixed length string known in literature as *genotype*, or alternatively *chromosome*. Several strategies have been proposed for the initialization; a classical one is to randomly draw the population. Afterward, the fitness function has to be evaluated for each element of the population. The identification of a suitable objective function, able to give a measure of the goodness of an element, is usually problem-dependent.

Generation creates a new population through two steps: selection and recombination. Selection draws an intermediate population where the recombination has to be applied. A common approach is to pick up elements, belonging to the actual population, proportionate to their fitness. Recombination acts on this intermediate population to generate a new one. Probabilistic transition operators crossover and mutation, are usually applied. In particular, crossover selects two elements from the intermediate population and creates a new element swapping a portion of their chromosomes with respect to a crossover point. On the other hand, mutation selects an element from the intermediate population and creates a new element modifying some bits of its chromosome.

3 The Clustered Evolutionary Monte Carlo Filter

The *Clustered Evolutionary Monte Carlo* filter (CE-MCL) has been conceived following the classical Sequential Monte Carlo filter schema mentioned above. The algorithm works on two different levels:

- Local level
- Global level.

At local level the algorithm finds out local maxima within each cluster, whereas at global level the best hypothesis is obtained by a comparison among the optimal solutions provided by each cluster.

In order to realize such behavior, the algorithm introduces two strategies:

- A dynamical clustering at resampling step
- An evolutionary action at each time-step.

3.1 Dynamical Clustering

The dynamical clustering provides a collection of particles subset that represents the best partition of the environment and where the probability to find out the real robot location is higher. Cluster identification is performed by means of the DBscan algorithm, which relies on a density-based notion of clusters [25]. Such algorithm offers several good properties, such as the ability of finding out clusters of arbitrary shapes, the advantage of collecting the *noisy* points, and an acceptable computational complexity. In particular, the possibility of collecting all the points belonging to any clusters turns out to be very useful in this context. In fact, it can be viewed as another mean to improve the diversity among particles. Moreover, in order to guarantee both a minimal coverage of the environment and further mitigate the degeneracy problem, a random action is introduced along with the dynamical clustering at resampling step. Such action reduces the similarity among particles randomly drawing a percentage of new samples.

3.2 Evolutionary Action

The evolutionary action, instead, is introduced to quickly find out local maxima within each cluster. From a genetic point of view a cluster represents the population, while the state space vector is the encoding string, e.g., the chromosome. The model of the sensor $P(z_k \mid x_k)$ is adopted as fitness function. This choice makes the local maxima to be prominent candidates to localize the robot, being the $P(z_k \mid x_k)$ part of the importance weight formulation as well. The evolutionary action is based on the probabilistic operators:

- Mutation
- Crossover.

Mutation creates a fixed percentage of new particles sampling from a circular area centered on the selected chromosome (Figure 1), whose radius is defined as follow:

$$\rho_k = \frac{1}{\sqrt{w_k^{(i)}}}. \tag{14}$$

The idea of defining the radius as an inverse function of the importance weight reflects the consideration that particles, with a considerable importance weight, should be located closer to the real robot than the others. Therefore, filling this area should be promising for a quicker localization.

On the other hand, crossover creates a fixed percentage of new particles combining chromosomes belonging to the same cluster. The idea of selecting parents within the same subset avoids unlikely recombination to happen, being clusters spatially organized.

At the end, the Clustered Evolutionary Monte Carlo Filter, taking advantage of these strategies, is able to both globally localize the real robot location and solve the kidnap problem, as well as to maintain the multi-hypotheses.

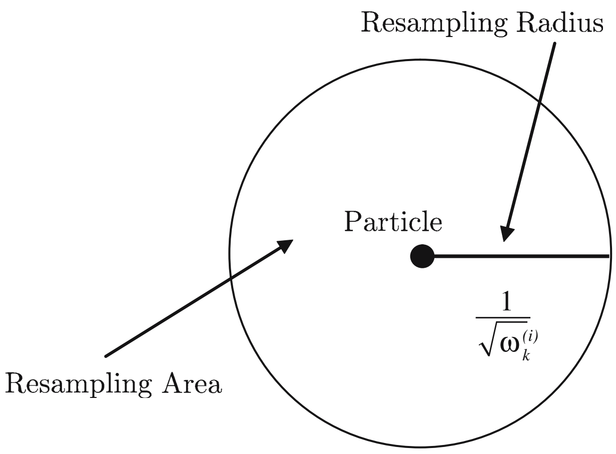


Figure 1 Choice of resampling area for mutation.

The Algorithm 2 shows a possible implementation schema for *The Clustered Evolutionary Monte Carlo Filter*.

Algorithm 2: Clustered Evolutionary Monte Carlo Filter

```
Data: $Bel^+(x_{k-1}) = \bigcup_{j=1}^{N_{clust}} \left\{x_{k-1}^{(i)}, w_{k-1}^{(i)}\right\}_j, u_{k-1}, z_k$
Result: $Bel^+(x_k)$
/* Importance Sampling                                                    */
Compute $\pi(x_k \mid d_{k-1}) = p(x_k \mid x_{k-1}, u_{k-1}) \cdot Bel^+(x_{k-1})$
for $i = 1$ to $Ns$ do
    Sample $\tilde{x}_k^{(i)} \sim \pi(x_k \mid d_{k-1})$
    Evaluate $w_k^{*(i)} = w_{k-1}^{*(i)} \cdot \frac{p(z_k|\tilde{x}_k^{(i)}) Bel^-(x_k)}{\pi(x_k|d_{k-1})}$
end
/* Normalization                                                          */
for $j = 1$ to $Ns$ do $\tilde{w}_k^{(i)} = \frac{w_k^{-(i)}}{\sum_{j=1}^{Ns} w_k^{*(i)}}$
/* Evolutionary Action                                                    */
for $j = 1$ to $N_{clust}$ do
    $\{\bar{x}_k^{(i)}, \bar{w}_k^{(i)}\}_j \leftarrow \begin{cases} Mutation(\{\tilde{x}_k^{(i)}, \tilde{w}_k^{(i)}\}_j) \\ Crossover(\{\tilde{x}_k^{(i)}, \tilde{w}_k^{(i)}\}_j) \end{cases}$
end
Evaluate $N_{eff} = \frac{1}{\sum_{i=1}^{Ns} (\bar{w}_k^{(i)})^2}$
/* Degeneracy Test                                                        */
if $N_{eff} \geq N_{thres}$ then
    $[\{x_k^{(i)}, w_k^{(i)}\}_j, N_{clust}] = [\{\bar{x}_k^{(i)}, \bar{w}_k^{(i)}\}_j, N_{clust}]$
else
    /* Resampling                                                         */
    /* 1° Step: Random action                                             */
    $\{\hat{x}_k^{(i)}, \hat{w}_k^{(i)}\} \leftarrow Random(\{\bar{x}_k^{(i)}, \bar{w}_k^{(i)}\})$
    /* 2° Step: Re-Clustering                                             */
    $[\{x_k^{(i)}, w_k^{(i)}\}_j, N_{clust}] \leftarrow DBScan(\{\hat{x}_k^{(i)}, \hat{w}_k^{(i)}\})$
end
$Bel^+(x_k) = \bigcup_{j=1}^{N_{clust}} \{x_k^{(i)}, w_k^{(i)}\}_j$;
```

3.3 Computational Complexity

In order to evauate the computational complexity of the algorithm, several analyses have been performed. A detailed theoretical study has been done along with an empirical validation of the obtained results. According to such a study, two different cases have to be taken into account:

- *simple* step
- resampling occurence.

In the first case, when the resampling is not considered, the complexity of the algorithm turns out to be $O(N_s M)$, where N_s is the number of particles and M is the number of segments describing the environment. Conversely, when the resampling occurs, the DBscan effort has to be considered. In this case the overall complexity of the algorithm is given by $max\{O(N_s log(N_s)),\ O(N_s M)\}$, where $O(N_s log(N_s))$ is the computational load of the DBscan [25]. Two remarks are now in order: the first one is that the number of segments (M) is usually at least one order of magnitude smaller than the number of particles (N_s); the second is that the resampling step, during a typical execution, takes less than 10% of the overall number of iterations. Therefore, it is correct to state that the *real* complexity that should be considered is the one of the simple step: $O(N_s M)$.

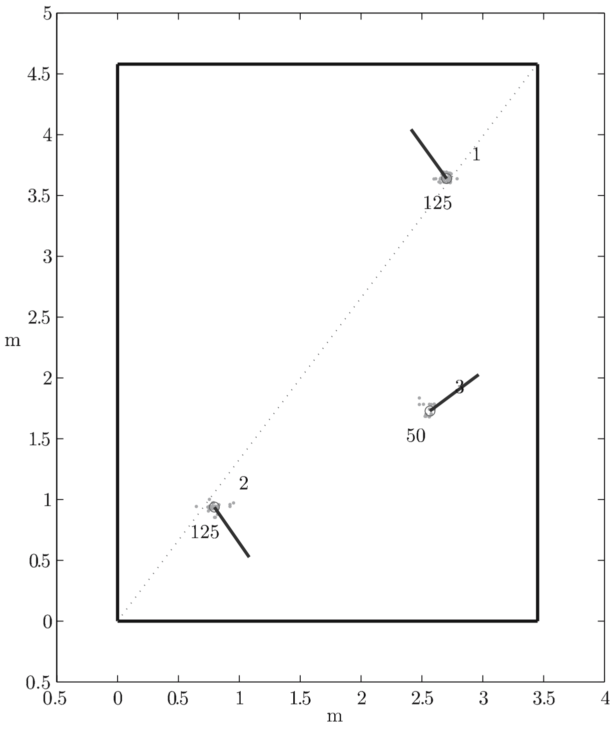
4 Experimental Results

The proposed algorithm has been tested in both simulated context and with real data in order to validate its capability to deal with the global localization problem. Several aspects have been thoroughly investigated. Particular attention has been paid to give evidence of the ability to carry on multi-hypotheses as well as to prove the ability to re-localize the robot when a kidnap occurs.

Simulated experiments have been performed using a framework developed under Matlab, which is able to provide a complete virtual environment. Real experiments have been executed on the mobile platform ATRV-Jr (All Terrain Robot Vehicle Junior) manufactured by iRobot. It is a skid steering vehicle mainly designed to operate in outdoor environments. The ATRV-Jr has four wheels differentially driven by two DC motors: the motion is achieved by a differential thrust on the wheel pairs at the opposite sides. The mobile robot is equipped with 17 sonar rangefinders, a laser scanner (Sick LMS-220), an inertial platform (Crossbow DMU-6X), and a GPS receiver (Garmin GPS35-HVS). The sensory system is connected to the ATRV-Jr's on board PC (Pentium II, 350 MHz) running Linux, through serial port on a Rockeport multiserial port card. The robot is delivered with a software development environment called MOBILITY, which provides full access to the software servers available on the mobile platform. In particular, each server is assigned to control a specific hardware component (sensors and actuators). In this way all of them are reachable from the network exploiting a CORBA interface.

Two different types of analysis have been performed. The first one as demonstrated the global algorithm aptitude to localize the robot as well as to carry on multi-hypotheses. The second more specialized analysis has proved the local algorithm ability to converge within each cluster.

Figure 2 Rectangular environment: CE-MCL iteration.



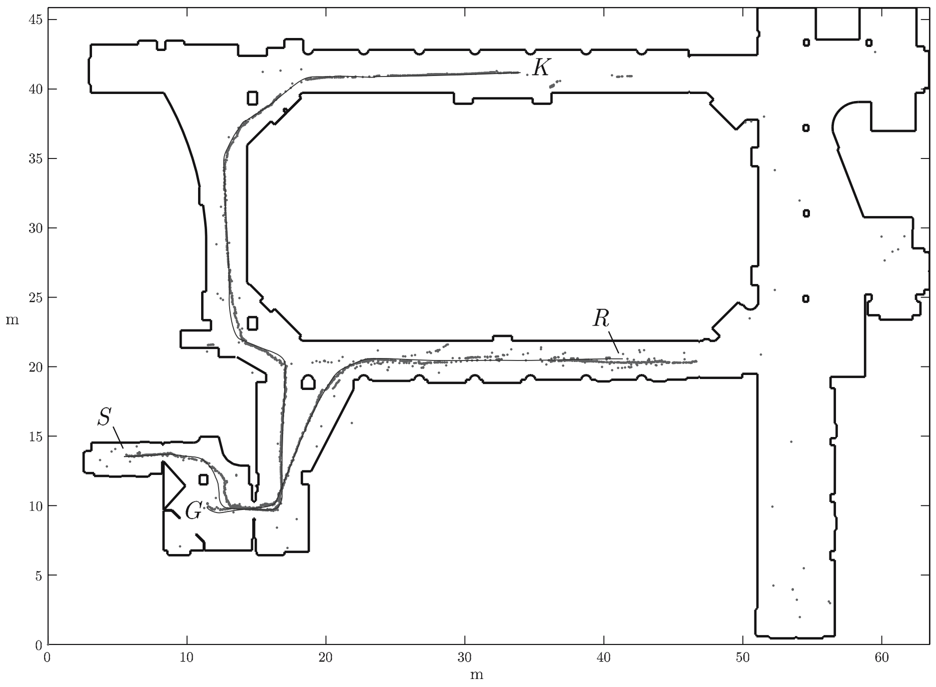


Figure 3 Complex environment: CE-MCL estimation result.

Two indexes of quality have been chosen to evaluate the correctness of the algorithm: the percentage of estimation failures and the entropy measurement of the particle set. The first one gives information about the reliability and the accuracy of the solution, the second one, coming from the *information theory* [26], provides a measurement of the uncertainty for a given random, and it is defined as:

$$H(X) = \sum_{i=1}^{n} p_i \, log_2(\frac{1}{p_i})$$

where, p_i is the probability of the ith outcome for a given event x. In this context, entropy can be properly applied to give an evaluation of the persistence of the diversity among particles.

For the first analysis, several environments have been taken into account to investigate each aspect of interest. In particular, a simple rectangular environment,shown in Figure 2, has been used to prove the ability of the algorithm to carry on

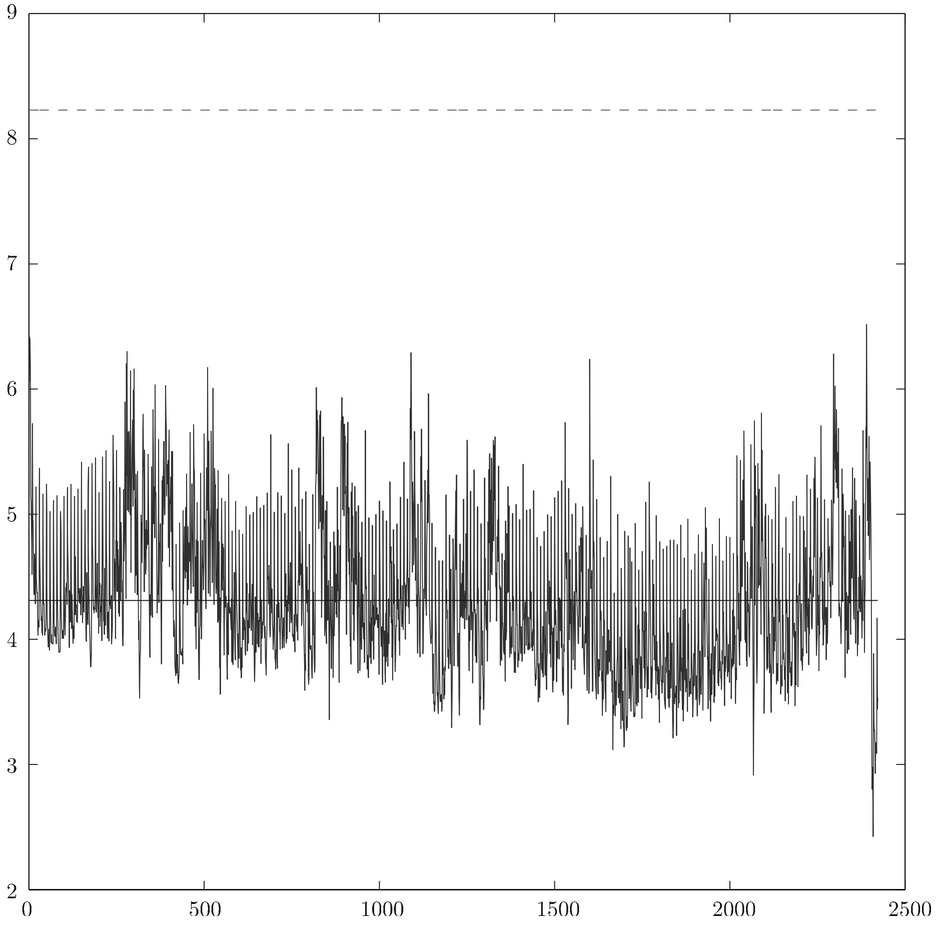


Figure 4 Complex environment: CE-MCL entropy measurement.

multi-hypotheses; a complex environment, shown in Figure 3, has been exploited to test the algorithm aptitude to solve both the global localization problem and the kidnap problem; finally, a T-shaped environment, shown in Figure 5, containing many glass elements, has been adopted to prove the algorithm robustness.

Figure 2 shows a typical CE-MCL iteration step for the rectangular environment. Points (green) represent particles, whereas circles (red) are located at the center of the mass of each cluster and, segments (blue) describe the mean orientation for all particles within each cluster. Due to the environmental symmetries, at each time-step at least two subset of hypotheses are maintained, in particular the ones located at the extreme of a segment splitting the environment in two equal parts. Further, such behavior seems to be reasonable as sensor data support both locations, nothing that the laser beam range is 8 m.

Figure 3 shows a typical CE-MCL estimation result for the complex environment previously mentioned. Points (red) represent the most likely hypotheses at each time

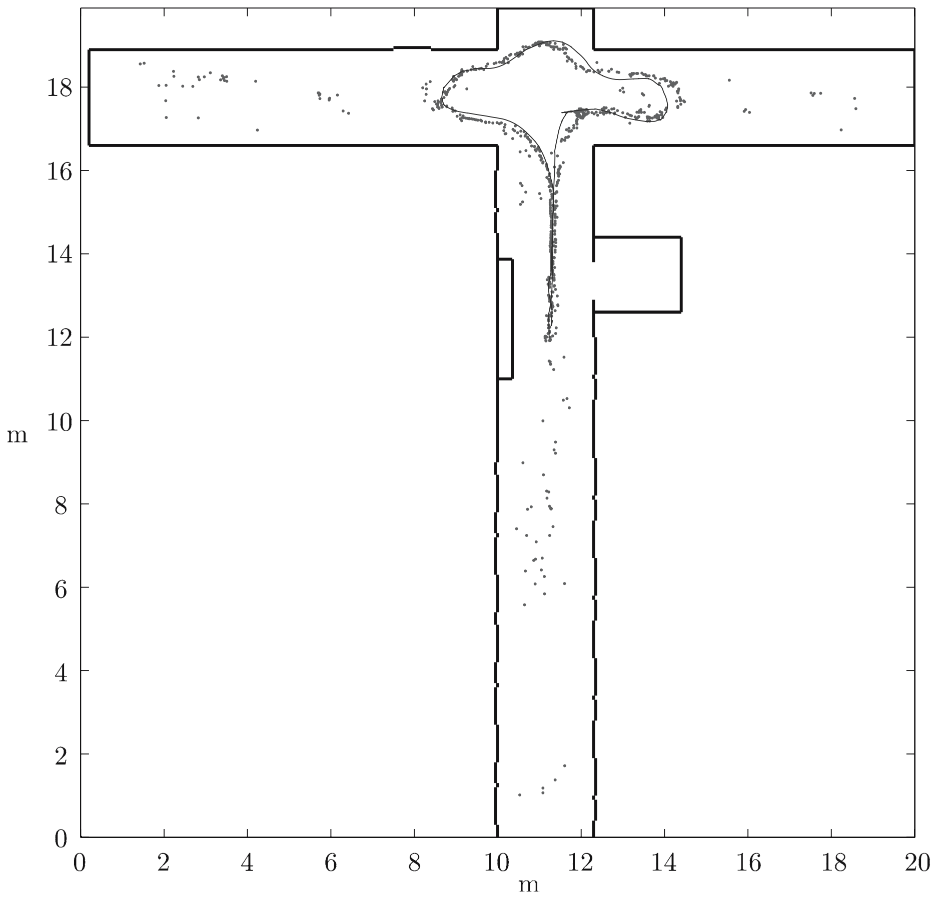


Figure 5 T-shaped environment: CE-MCL estimation result.

step, whereas the line (blue) represents the real robot path. In particular, *S* is the robot start point, *K* is the location at which the robot is kidnapped, *R* is the start point after the kidnap and finally, *G* represents the goal point of the robot.

The algorithm has been able to find out a rough estimation of the robot path without any knowledge about the starting robot location. Moreover, Figure 3 evidences the algorithm's ability to realize when a kidnap occurs, thus re-localize the robot. The remaining noisy points, consequence of a temporary bad estimation, prove the algorithm's tendency to explore the whole environment as well as to carry on the multi-hypotheses at each time step. Further, they might be easily removed, for instance relying on the kinematic model constraints of the mobile robot.

The algorithm has been runned several times in this environment to estimate the percentage of failures; the mean value settles around 30%, while the variance is ±4%.

Figure 4 shows the measurement of entropy for the experiment mentioned above. The red line represents the theoretical maximum entropy for the given set of

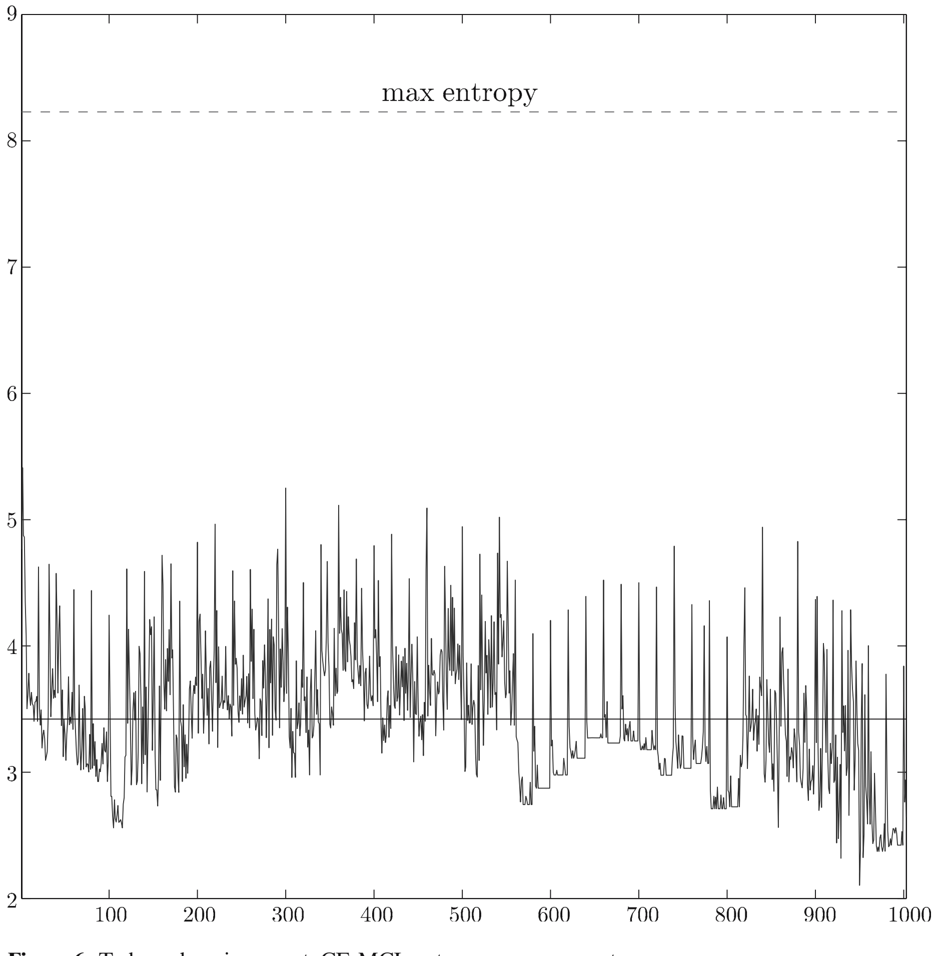


Figure 6 T-shaped environment: CE-MCL entropy measurement.

particles, whereas the blue line describes the entropy during the algorithm execution, and the black line is its mean value. In order to maintain the diversity among the particle set, such value should be high. However, the algorithm should also be able to converge to the real robot location. For the CE-MCL algorithm, the mean value settles around an intermedian value, giving evidence of the algorithm's aptitude to balance both needs.

To evaluate the algorithm robustness, an environment containing many glass elements and an evident structural ambiguity has been considered. Figure 5 shows a typical CE-MCL estimation result for such environment. The presence of structural ambiguities along with the noisiness of laser readings, due to the nature of glass, make the localization problem more difficult. Despite the fact that the accuracy of the estimation is lower than the previous experiment, and the percentage of failure is markedly higher (mean value 40%, variance ±5%) the algorithm has been able to localize the robot, even under these critical conditions.

Figure 6 shows the measurement of entropy for this environment. As in the previous experiment, the value settles around the median value, proving the algorithm's ability to maintain the diversity among the particles set.

The second type of analysis has been performed at cluster-level to figure out the local algorithm's behavior. Two aspects of interest have been considered to study the

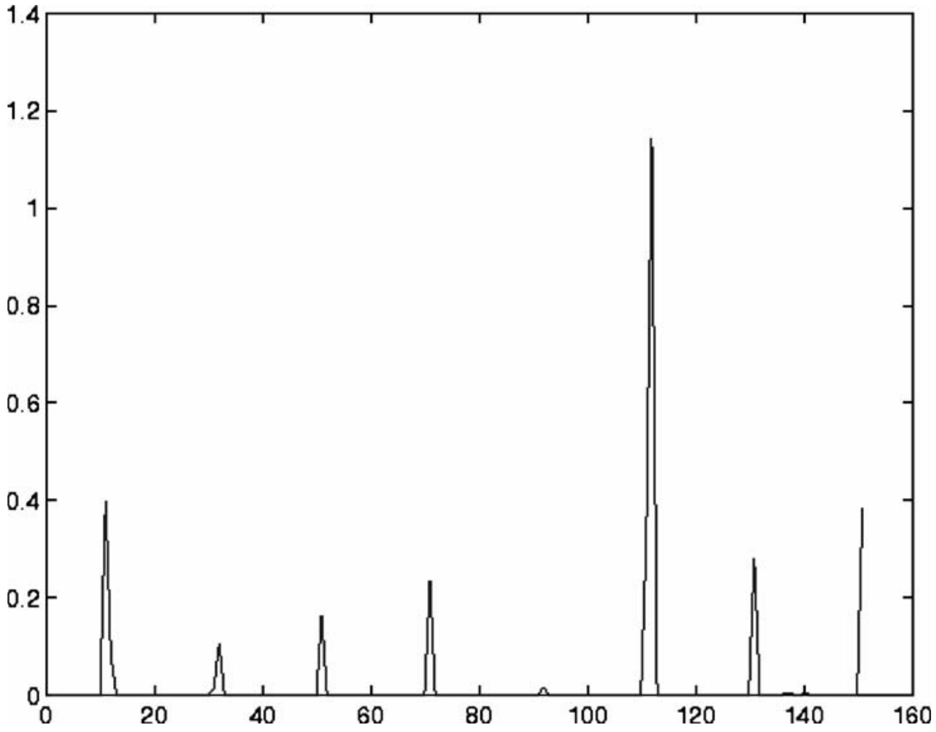


Figure 7 Corridor-like environment: CE-MCL variance.

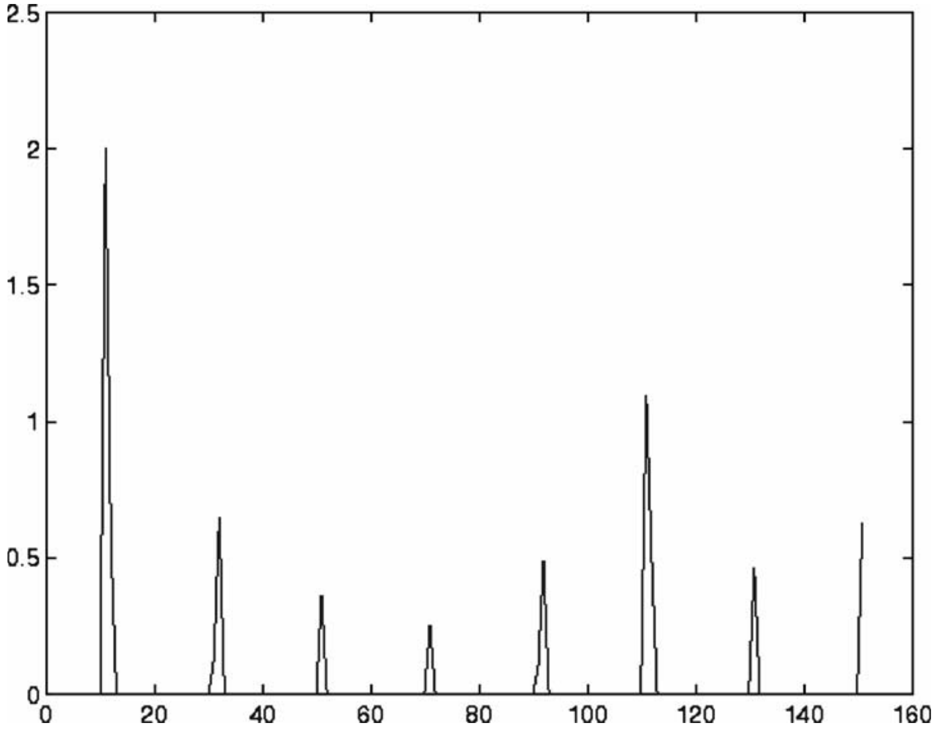


Figure 8 Corridor-like environment: CE-MCL entropy measurement.

convergence of particles within each cluster: a measure of similarity and the state variables variance.

Figure 9 shows a sequence of CE-MCL iterations for an additional regular environment exploited for such analysis. This sequence of images describes the algorithm behavior between two resampling steps. In particular, when the first resampling occurs (*a*), the algorithm recognizes six clusters (the last one is the collection of noisy points) with a visible dispersion for the elements within the environment. The following iterations point out the CE-MCL tendency to centralize the hypotheses around the center of mass of each cluster. Moreover, after few time-steps clusters are coarsely located along a line crossing the corridor. This deployment underlines the algorithm tendency to maintain only the most likely hypotheses after a full exploration of the environment.

From a mathematical standpoint, both the state space variables analysis and the measurement of entropy give evidence of these considerations. Figure 7 shows a typical variance trend within a cluster for all three state variables: peaks indicate the resampling effect, whereas slopes give evidence of the algorithm aptitude to make particles converge within each cluster. Figure 8 exhibits a typical measurement of entropy within a cluster: the trend is similar to the previous one due to the relationship among these concepts, when restricted to a single cluster.

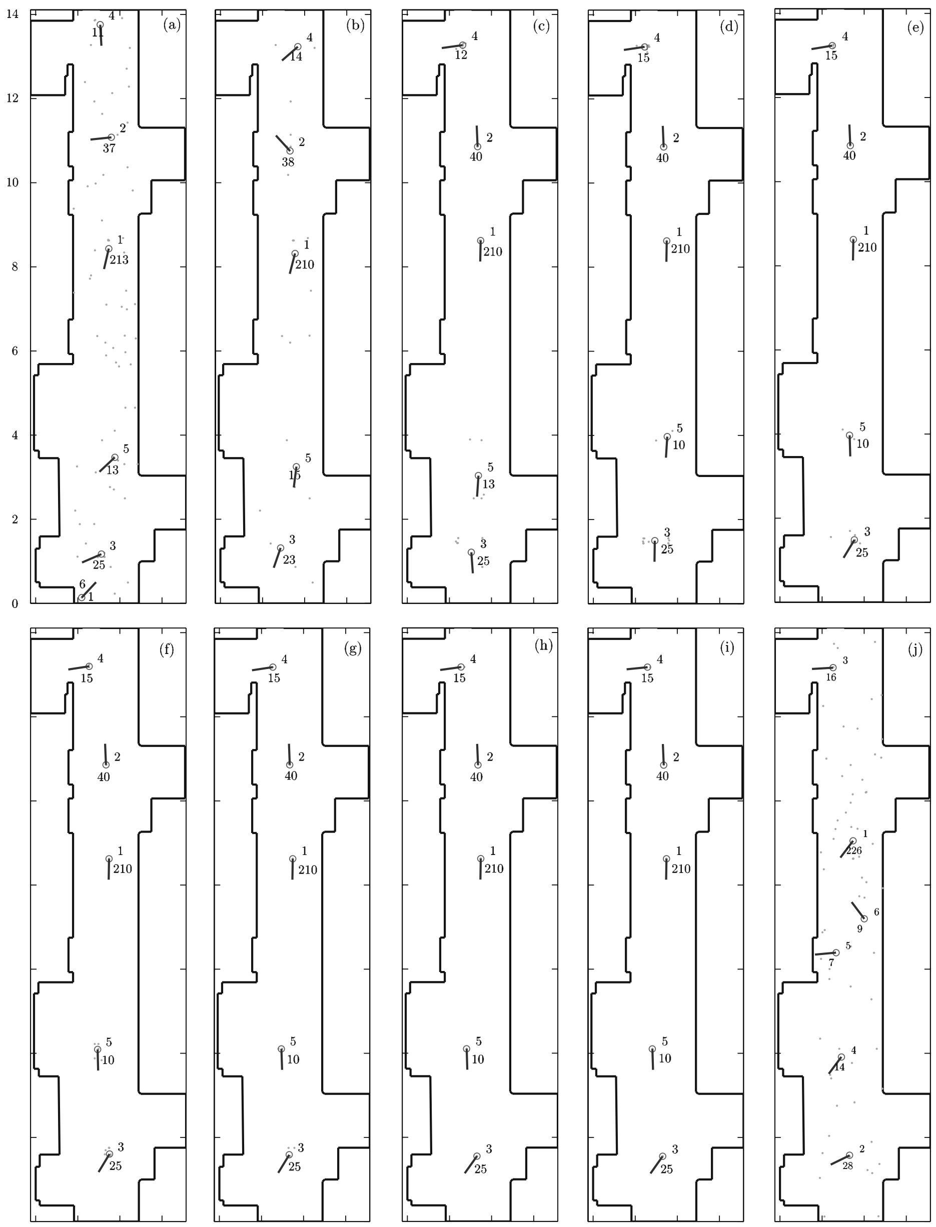


Figure 9 Corridor-like environment: CE-MCL sequence of iterations.

5 Conclusions

In this paper a new Monte Carlo Filter has been proposed to deal with the localization problem: the *Clustered Evolutionary Monte Carlo Filter* (CE-MCL). Such algorithm has been conceived to overcome the classical Monte Carlo Filter drawbacks. This goal has been achieved taking advantage of an evolutionary approach and a clustering method. In particular, the former has been exploited to quickly find out local maxima, whereas the latter, being dynamical, helps to obtain an effective exploration of the environment. The ability to provide a smart partition set of the research space along with the guarantee to converge within each subset, make the algorithm able to solve the localization problem and maintain the multi-hypotheses.

Note that the combined use of cluster+genetic offers several interesting advantages. At local level, being the size of research space smaller, the localization of the best solution is faster and the probability to stall on a suboptimal solution is lower. At global level, being the clustering dynamical and data-driven, an implicit parallelization of the research is possible and a better coverage of the environment is obtained, focusing the attention where the probability to find out the real robot pose is higher.

Exhaustive analyses have been performed on the robot ATRV-Jr manufactured by the IRobot, with the employment of several environments, to prove the effectiveness of the proposed algorithm. In particular, two different kinds of experiments have been considered: the first one has proved the algorithm ability to solve the global localization problem, even when a kidnap occurs; the second one has given evidence of the algorithm tendency to converge within each cluster and to guarantee an efficient exploration of the environment. Such analyses have shown the important role of the dynamical spatial clustering to provide an effective partion of the research space on which apply the evolutionary action. Therefore, the CE-MCL can find out local-maxima, guarantee a convergence to the most likely hypotheses, then maintain the diversity among particles and localize the robot.

Some interesting challenges remain for future works. For instance, an extension of the CE-MCL for the multi-robot scenario could be devised exploiting the cluster-based approach to better distinguish hypotheses. From a computational point of view, a dynamical number of particles could be introduced, to reduce the complexity of the algorithm.

References

1. Kalman, R.: A new approach to linear filtering and prediction problems. Trans. ASME J. Basic Eng. **82**, 35–44 (1960)
2. Maybeck, P.S.: Stochastics Models, Estimation and Control, vol. 1. Academic Press, New York (1979)
3. Panzieri, S., Pascucci, F., Ulivi, G.: An outdoor navigation system using gps and inertial platform. IEEE/ASME Trans. Mechatron. **7**(2), 134–142 (2002)
4. Schiele, B., Crowley, J.L.: A comparison of position estimation tecnhiques using occupancy grids. In: Proceedings of the IEEE International Conference on Robotics and Automation, San Diego, California, pp. 1628–1634 May 1994
5. Burgard, W., Fox, D., Hanning, D., Schmidt, T.: Estimating the absolute position of a mobile robot using position probability grids. In: Proceedings of the 13th National Conference on Artificial Intelligence, Portland, pp. 896–901, 1996
6. Burgard, W., Derr, A., Fox, D., Cremers, A.B.: Integrating global position estimation and position tracking for mobile robots: the dynamic markov localization approach. In: Proceedings of the International Conference on Intelligent Robot and Systems, Victoria, Canada, 1998
7. Doucet, A.: On sequential simulation-based methods for bayesian filtering, CUED/F-INFENG/TR.310. Department of Engineering, Signal Processing Group, Tech. Rep., Cambridge University Press, UK (1998)

8. Handschin, J., Mayne, D.: Monte carlo techniques to estimate the conditional expectation in multi-stage non-linear filtering. Int. J. Contemp. **9**(5), 547–559 (1969)
9. Handschin, J.: Monte carlo techniques for prediction and filtering on non-linear stochastic processes. Automatica **6**, 555–563 (1970)
10. Akashi, H., Kumamoto, H.: State estimation for systems under measurements noise with markov dependent statistical property–an algorithm based on random sampling. In: Proceedings of the 6th Conference IFAC, 1975
11. Isard, M., Blake, A.: Condensation–conditional density propagation for visual tracking. Int. J. Comput. Vis. **29**(1), 5–28 (1998)
12. Wang, X., Chen, R., Liu, J.: Monte carlo bayesian signal processing for wireless communications. J. VLSI Signal Process. **30**(30), 89–105 (2002)
13. Fox, D., Burgard, W., Dellaert, F., Thrun, S.: Monte carlo localization: efficient position estimation for mobile robots. In: Proceedings of the 16th National Conference on Artificial Intelligence (AAAI'99), AAAI/MIT, Menlo Park, California, 1999
14. Arulampalam, M.S., Maskell, S., Gordon, N., Clapp, T.: A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. IEEE Trans. Signal Process. **50**(2), 178–188 (2002)
15. Liu, J., Chen, R.: Sequential monte carlo methods for dynamical systems. J. Am. Stat. Assoc. **93**, 1032–1044 (1998)
16. Zaritskii, V., Svetnik, V., Shimelevich, L.: Monte carlo techinique in problems of optimal data processing. Autom. Remote Control **12**, 95–103 (1975)
17. Milstein, A., Sánchez, J., Williamson, E.: Robust global localization using clustered particles filtering. In: Proceedings of the 18th National Conference on Artificial Intelligence, Edmonton, Alberta, Canada, 2002
18. Kwok, N.M., Fang, G., Zhou, W.: Evolutionary particle filter: re-sampling from the genetic algorithm perspectve. In: IEEE/RSJ International Conference on IROS, Sydney, Australia 2005
19. Gasparri, A., Panzieri, S., Pascucci, F., Ulivi, G.: Genetic approach for a localisation problem based upon particle filters. In: Proc. of 8th International Symposium on Robot Control (SYROCO2006), Bologna, Italy (2006).
20. Fox, D.: Markov localization: A probabilistic framework for mobile robot localization and navigation. PhD dissertation, Diss, University of Bonn, Germany (1998)
21. Handshin, J.: Monte carlo techniques for prediction and filtering of non-linear stochastic processes. Automatica **6**, 555–563 (1970)
22. Holland, J.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, Michigan (1975)
23. Darwin, C.: On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life. J. Murray, London, UK (1859)
24. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, Massachusetts (1989)
25. Ester, M., Kriegel, H.P., Sander, J., Xiaowei, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), AAAI, Portland, Oregon, pp. 226–331, 1996
26. Shannon, C.: A mathematical theory of communication. Bell Syst. Tech. J. **27**, 379–423, 623–656 (1948)