

Efficient Active Global Localization for Mobile Robots Operating in Large and Cooperative Environments

Andreu Corominas Murtra¹, Josep M. Mirats Tur¹, Alberto Sanfeliu^{1,2}

Abstract—This paper presents a novel and efficient framework to the active map-based global localization problem for mobile robots operating in large and cooperative environments. The paper proposes a rational criteria to select the action that minimizes the expected number of remaining position hypotheses, for the single robot case and for the cooperative case, where the lost robot takes advantage of observations coming from a sensor network deployed on the environment or from other localized robots. Efficiency in time complexity is achieved thanks to reasoning in terms of the number of hypotheses instead of in terms of the belief function. Simulation results in a real outdoor environment of $10.000m^2$ are presented validating the presented approach and showing different behaviours for the single robot case and for the cooperative one.

I. INTRODUCTION

Mobile robot map-based global localization is the problem of estimating the position robot in the map reference frame $(x_r^m, y_r^m, \theta_r^m)$, for the general case where no initial guess for the robot position is given and only the robot map and the real-time data from the robot onboard sensors are available. Mobile robotics community has agreed to point out global localization as a key topic when designing autonomous systems [1]. Since GPS-based systems have been considered neither enough robust nor accurate for mobile robotics [2], [3], alternative or complementary methods have to be designed, specially in urban environments where radioelectric shadowing and multipath propagation corrupt the GPS signal.

Uncertainty along the state space in map-based global localization methods is usually represented with a multi-gaussian distribution [4], [5], [3], a complete discretization of the state space [6] or by means of a set of weighted particles sampling the state space [7], [8]. In any case, the distribution is considered multi-peak for robust solutions and, therefore, an active strategy has to send commands to the robot, in robot coordinate frame since the robot is not yet located, driving the robot to other places of the environment, with the aim of disambiguate the multi-hypothesis situation. Figure 1 shows the multi-hypothesis situation when a particle filter is executed in a real environment of $10.000m^2$.

Existing methods for active global localization can be divided in heuristic, geometric and entropy-based approaches.

Examples of *heuristic* strategies are proposed by [4], [9]. The former describes a set of rules that command the robot to the non visited areas where more features

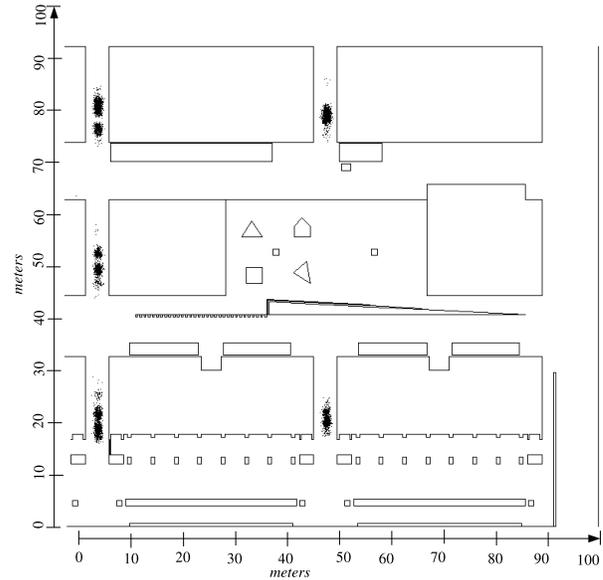


Fig. 1. Usual multi-hypothesis situation in global localization after some iterations of a particle filter in an environment of $10.000m^2$

are expected to be found, while the later drives the robot to the nearest obstacle found in the map, taking into the account all position hypotheses. Heuristic approaches lead to generalization problems when they are exported to dynamic and large environments with other robots operating around.

A second trend in active strategies are the geometric approaches [10], [11]. In [10] authors address the problem of localizing a lost robot executing the minimum distance path by means of a random search over a world overlapping local maps of all hypotheses. More recently, [11] proposes a minimalist approach using only odometry data to actively localize a robot up to the symmetries of the environment. Both works suppose deterministic sensor readings and robot motions, and shows simulated experiments in an ideal polygonal world. Exportation of these approaches to real, dynamic and cooperative environments remains an open question.

A third family of active techniques, called *entropy-based*, addresses the uncertainty present in mobile robotics. In [12] a general formulation is presented, based on the Markov framework of the same authors [6]. The approach evaluates a finite set of actions computing the expected future *belief* and its entropy associated, future in the sense that each action was executed. The strategy selects the action that minimizes the expected entropy, therefore selecting the action that most concentrate the expected position distribution. The complete

Work under foundation grant of URUS IST-045062 research program.

¹ Institut de Robòtica i Informàtica Industrial, IRI (UPC-CSIC). C/Llorens i Artigas, 4-6, 2nd floor. Barcelona, Spain. www-iri.upc.es

² Universitat Politècnica de Catalunya, UPC. Barcelona, Spain.

acorominas@iri.upc.edu, jmirats@iri.upc.edu, asanfeliu@iri.upc.edu

discretization of the state space provokes high computational complexity, critical in large environments. Authors in [13] implements an entropy-based active localization algorithm, but using a stereo camera and a particle filter framework that improves the computational cost. However, computational efforts remain hard and in the practical situation authors have to reduce drastically the set of actions and the size of the environment to obtain a reasonable computation delay.

The present work proposes a *novel* and *efficient* solution to the active map-based global localization problem for mobile robots operating in large and cooperative environments. Since the goal application is to navigate in environments such as urban settings, with a group of robots operating on it and/or with a deployed sensor network [14], both computational efficiency and cooperation are addressed. Novelty is thanks to consider a cooperative environment in the action selection for global localization. Efficiency is thanks to reduce the time complexity for the active strategy, since computing is performed in terms of the number of position hypotheses, instead of in terms of the whole belief function. After some definitions (section II), section III formulates the active strategy in a general way for both, the single robot case and the cooperative one. The paper discusses an implementation of the proposed active method in section IV. Section V studies the computational complexity comparing the presented approach with the entropy-based ones. Finally, simulation results in a real environment of $10.000m^2$, are presented in section VI, showing different behaviours for the single robot case and for the cooperative one.

II. PROBLEM STATEMENT AND DEFINITIONS

The presented active strategy starts assuming some considerations. Please note that an implementation of these listed considerations is discussed in section IV.

- The robot has a *map*, \mathcal{M} , as a data base describing the area where it operates following an environment model. A position p in the map frame is $X_p^m = (x_p^m, y_p^m, \theta_p^m)$.
- The robot can process up to N_O *real* observations coming from its onboard sensors or from remote observers such as other robots or sensors deployed in the environment. The n^{th} real observation at time t is denoted as o_n^t . A model for the n^{th} observation, $o_n^s(X_p^m)$, computes a *synthetic* observation, given the robot state X_p^m and the map \mathcal{M} , as the expected n^{th} observation if the robot was at $(x_p^m, y_p^m, \theta_p^m)$. Both o_n^t and o_n^s are in the observation space O_n .
- A likelihood function $L_n(\cdot)$ is defined that calculates the matching between two n^{th} observations being real or synthetic. This function approaches to 1 for similar observations and goes close to 0 for distinctive ones.

$$L_n(o_n^{t,s}, o_n^{t,s}) \in [0, 1] \quad (1)$$

- The conditional probability of a real observation given the robot is at state X_p^m , $p(o_n^t|X_p^m)$, is computed with the observation model $o_n^s(X_p^m)$ and the likelihood function $L_n(\cdot)$ as:

$$p(o_n^t|X_p^m) = L_n(o_n^t, o_n^s(X_p^m)) \quad (2)$$

- This probability can be also computed for a synthetic observation, indicating how distinctive is the position X_q^m to the position X_p^m from the point of view of the n^{th} observation. This fact is the core of the herein proposed active strategy and is formally defined as:

$$p(o_n^s(X_q^m)|X_p^m) = L_n(o_n^s(X_q^m), o_n^s(X_p^m)) \quad (3)$$

- A set of N_H position hypotheses for the robot is defined as $H = \{h_1, \dots, h_{N_H}\}$ where the i^{th} hypothesis is defined with a position in the map frame, $X_{h_i}^m = (x_{h_i}^m, y_{h_i}^m, \theta_{h_i}^m)$, a covariance matrix, C_{h_i} , and a probability associated, p_{h_i} :

$$h_i = \{X_{h_i}^m, C_{h_i}, p_{h_i}\}, \forall i = 1..N_H; \sum_{i=1}^{N_H} p_{h_i} = 1 \quad (4)$$

- Finally, a path planning technique is required as discussed in subsection III-A.2, to provide to the robot a collision free path to execute.

With all these basic assumptions, the problem definition of an active strategy is where to move the robot in order to reduce the hypotheses set. The proposed strategy wants to exploit the map, searching for positions where distinctive observations are expected among the hypotheses.

III. ACTIVE STRATEGY FRAMEWORK

The proposed framework is divided in three steps. The first one consists in randomly generate a set of *exploration particles* in the robot frame, as robot candidate destinations (candidate actions). The second step validates these exploration particles if a *multi-hypothesis path* exists between the robot and the given exploration particle. The third step computes, for each validated exploration particle, the *expected number of remaining hypotheses* given that the robot goes to that exploration particle. The exploration particle, as a position in the robot frame, with minimum expected number of remaining hypotheses is the selected one to drive the robot.

A. Single Robot Case

1) *Generating Exploration Particles*: Let's call the k -th exploration particle, ϵ_k^r , as a random position in the robot coordinate frame generated within a given disk of radius R_ϵ around the robot. R_ϵ is called the *exploration radius*.

$$\epsilon_k^r = X_{\epsilon_k}^r = (x_{\epsilon_k}^r, y_{\epsilon_k}^r, \theta_{\epsilon_k}^r) \quad (5)$$

Under the assumption that h_i is the true hypothesis, we can express ϵ_k^r in the map frame as:

$$\epsilon_{ki}^m = \epsilon_k^r | h_i = \begin{bmatrix} x_{h_i}^m \\ y_{h_i}^m \\ \theta_{h_i}^m \end{bmatrix} + \begin{bmatrix} \cos\theta_{h_i}^m & -\sin\theta_{h_i}^m & 0 \\ \sin\theta_{h_i}^m & \cos\theta_{h_i}^m & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{\epsilon_k}^r \\ y_{\epsilon_k}^r \\ \theta_{\epsilon_k}^r \end{bmatrix} \quad (6)$$

Equation 6 shows that a single exploration particle ϵ_k^r becomes a set of N_H positions in the map when it is translated to the map frame, since we have to consider all hypotheses and, therefore, translate ϵ_k^r for each hypotheses h_i , $\forall i \in [1..N_H]$. Fig. 2 illustrates this observation.

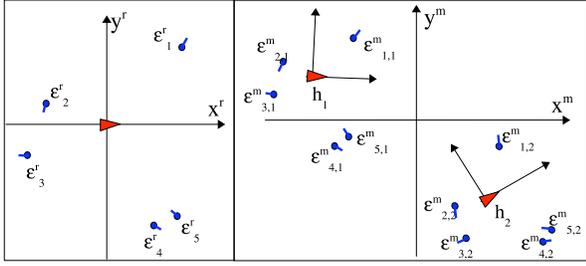


Fig. 2. A set of 5 exploration particles in the robot coordinate frame (left) and their transformation to map coordinate frame (right) when $N_H = 2$.

2) *Multi-hypothesis Path Planning*: Even if e_k^r is expressed in the robot frame and, therefore, the robot knows where the exploration particle is positioned, since e_k^r can be beyond the sensor horizons we have to assure that a free path exists between the robot and e_k^r for all hypotheses. We have called this step *multi-hypothesis path planning* (MHPP), as the planning of a path expressed in the robot frame taking into account all hypotheses constraints. Figure 3 draws the MHPP approach in an illustrative geometric world. For this step, C_{h_i} can be used as a clearance factor.

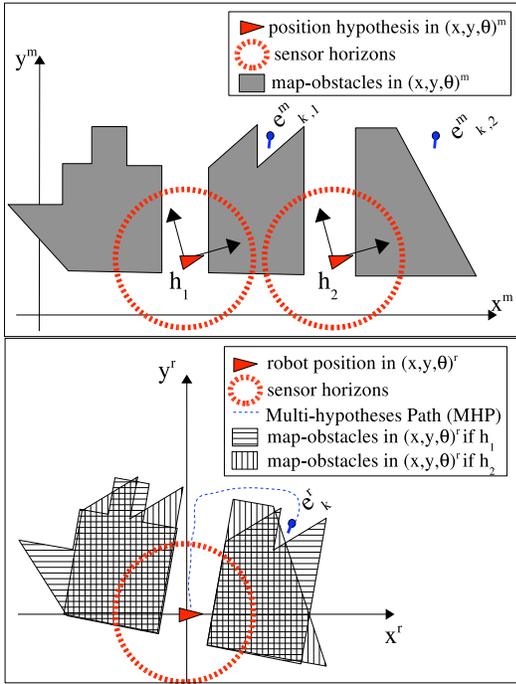


Fig. 3. Multi-hypothesis Path (MHP) in an illustrative geometric world. Map coordinate frame on the top and robot coordinate frame on the bottom.

If a multi-hypothesis path (MHP) exists between the robot and the e_k^r , we label e_k^r as a valid candidate destination, e_k^r , and we add it to the set of all valid exploration particles E . Summarizing, the output of the first and second steps will be a set E of N_E exploration particles that are connected to the robot with a MHP:

$$E = \{e_1^r, \dots, e_k^r, \dots, e_{N_E}^r\} \quad (7)$$

3) *Computing Hypotheses Reduction*: The goal of this third step is to compute $\hat{N}_H(e_k^r)$, as the expected number of remaining hypotheses given that the robot goes to e_k^r and senses the environment. To compute this number, we first must be able to compute $\hat{N}_H(e_k^r|h_i)$, as the expected number of remaining hypotheses if the robot moves to e_k^r given that h_i is the true position hypothesis. Using the observation model and the likelihood function discussed in section II, we formulate, when $N_O = 1$:

$$\hat{N}_H(e_k^r|h_i) = \sum_{j=1}^{N_H} p(o_1^s(e_{kj}^m)|e_{ki}^m) \quad (8)$$

When N_O observations are available, if we assume independency between them, equation 8 generalizes as:

$$\hat{N}_H(e_k^r|h_i) = \sum_{j=1}^{N_H} \prod_{n=1}^{N_O} p(o_n^s(e_{kj}^m)|e_{ki}^m) \quad (9)$$

We can now formalize the $\hat{N}_H(e_k^r)$ as the sum of each $\hat{N}_H(e_k^r|h_i)$ weighted by the probability of the i^{th} hypothesis being true, p_{h_i} :

$$\hat{N}_H(e_k^r) = \sum_{i=1}^{N_H} \hat{N}_H(e_k^r|h_i) \cdot p_{h_i} \quad (10)$$

Please note that $\hat{N}_H(e_k^r) \in [1, N_H]$ since $p(o_n^s(e_{kj}^m)|e_{ki}^m) \in [0, 1]$ as stated in section II. For an exploration particle e_k^r having similar synthetic observations $\forall h_i$, all the probabilities $p(o_n^s(e_{kj}^m)|e_{ki}^m)$ will be close to 1 and, therefore, $\hat{N}_H(e_k^r|h_i) \approx N_H$. Given the assumption of equation 4, $\hat{N}_H(e_k^r)$ will also result in $\approx N_H$. This case implies that the position of e_k^r has synthetic observations too similar for all position hypotheses, and, therefore, it is an exploration particle that will not disambiguate at all the situation. On the other hand, when an exploration particle has completely different synthetic observations $\forall h_i$, the probability $p(o_n^s(e_{kj}^m)|e_{ki}^m)$ will be close to zero $\forall i \neq j$, but it will take one for $i = j$. Again, given the assumption of equation 4, $\hat{N}_H(e_k^r) \approx 1$. In this case, the exploration particle e_k^r is expected to completely disambiguate the situation since all synthetic observations are entirely different for each h_i .

With this well delimited results, we can use the $\hat{N}_H(e_k^r)$ as the expected number of remaining hypotheses given that the robot goes to e_k^r , so the robot will start path execution driving itself to the position e_k^r with minimum $\hat{N}_H(e_k^r)$.

B. Cooperative Case

This subsection analyses the case of a lost robot which is a member of a network robot system. In this situation the active strategy selects, as in section III-A, one action exploiting the robot onboard sensors and the map, but also uses the potentialities of integrating remote observations. Let's define the coverage space of the sensor network as:

$$C_{CN} = \bigcup_{c=1}^{N_C} C_c, \quad C_{CN} \subset X^m \quad (11)$$

where C_c is the coverage area of the c^{th} sensor of the network and N_C the number of sensors of the network. We can also define the coverage space of the robots, which is time depending, as:

$$C_{RN}^t = \bigcup_{r=1}^{N_R} C_r^t, \quad C_{RN}^t \subset X^m \quad (12)$$

where C_r^t is the coverage area of the r^{th} robot at time t and N_R the number of robots. For a lost robot, $C_r^t = \emptyset$.

In the proposed network robot system, both C_{CN} and C_{RN}^t are data available on the central server, since it knows where the sensors are deployed and where the non lost robots are. A robot can request both coverage spaces at a given time and, thus a lost robot can use this data for local processing when it is executing the active global localization strategy.

In this context, the active strategy will be the same that the one exposed in subsection III. Evaluation of actions will be done by equations 9 and 10, but considering that the robot can use external observations done by other observers such as a camera network or well localized robots. In equation 9, and in order to consider remote observations for the active strategy, the lost robot has to evaluate if e_{kj}^m is in $C_{CN} \cup C_{RN}^t$. If this is the case, a remote observation for that position is available and the $p(o_n^s(e_{kj}^m)|e_{ki}^m)$ can be computed where o_n^s is the model for that remote observation.

The effect of this is that exploration particles expected to be in the coverage space, $C_{CN} \cup C_{RN}^t$, will be attractive to move the robot since disambiguation can be done via remote observations instead of only considering the robot exteroceptive observations. This is a situation of an active approach considering the potentialities of a cooperative environment, taking advantage of information sharing.

As an illustrative example, the GPS is a particular case of this cooperative context since the GPS satellite network acts as a sensor network. Assuming that we have a map of the GPS coverage in our environment, a lost robot equipped with a GPS receiver out of coverage will be attracted by actions driving the robot to areas where GPS is available.

IV. IMPLEMENTATION

This section briefly details the implementation of the proposed active approach theoretically formulated in section III. More details are given in [15].

1) *Environment Model*: The environment model is based on the vector format of GIS (Geographical Information System) [16]. The map represents a set of obstacles. Each obstacle is described as a set of straight segments and some semantic information related. Each segment is parametrized with two points in the map frame and also accompanied of some semantic and height information. To reduce the computational cost of the routines dealing with the map, each obstacle is enclosed in a minimum bounding ball [17]. The main interest of this GIS-based map is its compactness required for large environments, the possibility to perform localization and path planning on it and the use of an standard model instead of an *ad hoc* representation.

2) *Platform and Sensory System*: We simulate the pioneer 3pat platform (MobileRobots Inc.) equipped with wheel encoders, an electronic compass TCM2 (PNI corp.) and a laser scanner RS4 (Leuze corp.). Encoders are used as proprioceptive sensors for the position tracking task. The electronic compass serves a first coarse estimation of the robot orientation (θ_0^m). The laser scanner gives a set of $N_S = 133$ points over the scan aperture of $(-95, 95)$ degrees and the maximum laser range is limited to $RMAX = 12m$.

3) *Sensor Model*: The laser scanner is used as the onboard exteroceptive sensor for the proposed active approach. In order to compute the synthetic observations, $o_1^s(X_p^m)$, the ray tracing function is executed N_S times from a given position X_p^m sweeping the heading to cover all the scan aperture. A synthetic laser scanner observation computed from X_p^m is:

$$o_1^s(X_p^m) = (o_{1,1}^s(X_p^m), \dots, o_{1,N_S}^s(X_p^m)) \quad (13)$$

This $o_1^s(X_p^m)$ function is the explicit observation model considered for the proposed active strategy.

4) *Likelihood Function*: Given two synthetic observations $o_1^s(X_p^m)$ and $o_1^s(X_q^m)$, computed from $X_p^m \in X^m$ and $X_q^m \in X^m$, the likelihood function implemented is:

$$L_1[o_1^s(X_p^m), o_1^s(X_q^m)] = \frac{1}{N_S} \sum_{k=1}^{N_S} \text{erfc}\left(\frac{|o_{1,k}^s(X_p^m) - o_{1,k}^s(X_q^m)|}{\sigma_L \sqrt{2}}\right) \quad (14)$$

where $\text{erfc}()$ is the complementary error function and σ_L represents the uncertainty of the laser readings and that of the position hypotheses. The presented results uses $\sigma_L = 1.5m$.

5) *Global Localization with Particle Filtering*: We have implemented a particle filter as the previous step of the active strategy, following the well established framework of [7]. The filter initializes a set of N_P position particles, $s_i = \{X_{s_i}^m, w_{s_i}\}$ with $X_{s_i}^m \in X^m$ and $w_{s_i} \in [0, 1]$, sampling randomly the location space $(x, y)^m$, but sampling the θ^m space in an interval of 10 degrees around the first compass reading θ_0^m . The reliability of each particle with the current real laser observation, $p(o_1^t | X_{s_i}^m)$, is computed as $L_1[o_1^t, o_1^s(X_{s_i}^m)]$. A weight, as the product of this likelihood and the previous weight, is assigned to each particle. The *resampling step* samples the position space, according to the particle weights, so likely areas are successively more populated. Robot movements translate all particles by means of the robot motion model using the odometry data. The presented results are obtained with $N_P = 5000$.

6) *Clustering and Hypotheses Generation*: The implemented particle filter for global localization deals with multiple hypotheses in an implicit way as concentrations or peaks in the belief distribution (see figure 1). A clustering step converts the position particle set to a reduced hypotheses set ($N_H \ll N_P$) as the input of the proposed active strategy.

Clustering is implemented by means of a recursive routine that starts with the set of position particles ordered by their weights $w_{s_i} \geq w_{s_j}$ when $i < j$. Let K_k be a cluster and $c(K_k)$ its centroid. Initially, the routine creates the first cluster using the first particle $c(K_1) = (X_{s_1}^m)$. The rest of

the particles, $j = 2, \dots, N_P$, will join to an already created cluster if $d(X_{s_j}^m, c(K_k)) < K_{th}$ or, otherwise, will create a new cluster. K_{th} is the parameter fixing clustering size and $d()$ is the euclidean distance between two positions in the map. When a particle joins to an already created cluster, the centroid is updated using all the particles currently being in that cluster as a weighted mean of their positions. When clustering finishes, the covariance matrix for each cluster is computed and the set H of position hypotheses is generated.

7) *Multi-hypothesis Path Planning*: The Rapidly-Exploring Random Trees (RRT) approach has been implemented [18]. The tree is computed in the robot frame translating the map obstacles to this frame for each hypotheses in a similar way as equation 6 does. Exploration radius is set to $R_\epsilon = 20m$ and the number of exploration particles is $N_E = 30$.

8) *Cooperative Environment*: The cooperative case is simulated placing omnidirectional cameras in known positions of the environment. If a line of sight between the camera and a position $(X_p^m) \in X^m$ exists, we consider that the camera can detect and localize a robot in that position. The range of the camera detection is limited to $R_C = 8m$. This simple model can represent fixed networked cameras/sensors or well localized robots with omnidirectional detection capabilities.

V. COMPUTATIONAL COMPLEXITY

As equations 9 and 10 suggest, the time complexity to evaluate a single action in the proposed active strategy is $\mathcal{O}(N_H^2 \cdot N_O)$. Therefore, the time complexity of evaluating a set of N_A actions results on $\mathcal{O}(N_H^2 \cdot N_A \cdot N_O)$. In the particular case of the proposed implementation, since the observations are computed on-line, N_O becomes $N_S + N_C$, which refers to the laser scan points and the number of cameras respectively. In terms of memory complexity, the presented implementation is extremely efficient since the spatial representation is based on the GIS vector format and no sensor-appearance data is stored in the map database, thus avoiding space discretization and huge representations. The memory complexity of this map model has not been analyzed in this work but the real environment of about $10.000m^2$, used as a testbench area, is represented with a map of about $40K Bytes$, supposing a very efficient world representation.

In the work based on the Markov framework [6], time complexity behaves as $\mathcal{O}(N_X^2 \cdot N_A \cdot N_S)$, where N_X is the number of all possible position states, N_A the size of the action set and N_S the number of sensings at each state. In order to reduce the computational cost, authors precompute the sensor model and cluster the belief distribution, forming a set of N_{Xg} Gaussians in a runtime step, reducing time complexity to $\mathcal{O}(N_X \cdot N_{Xg} \cdot N_A)$, with $N_{Xg} \ll N_X$. This clustering step is similar to that performed by our proposed approach in the sense of creating a set of Gaussians instead of having a sampled belief distribution. Therefore, we can suppose that $N_{Xg} \sim N_H$. However, the term N_X remains in the time complexity expression for this approach. Due to the complete discretization of the state space, N_X grows up with the environment size being critical in large areas.

Using the same entropy-based approach but based on the particle representation of the uncertainty, the work presented on [13] has a time complexity of $\mathcal{O}(N_P^2 \cdot N_A \cdot N_J)$, where N_P is the size of the particle set, N_A the number of actions to be evaluated, and N_J an observation model parameter. Authors precompute the observation model, reducing the time complexity to $\mathcal{O}(N_P^2 \cdot N_A)$ but increasing the memory complexity since the precomputations have to be stored in an appearance map. Since $N_P \ll N_X$ is a general case, this work drastically reduces the time complexity in comparison with [6]. However, the complexity remains high, specially for large environments where the amount of particles needed to globalize the robot is also large. In the practical experimentation, authors in [13] report the requirement to reduce the action set and the size of the environment.

Table I summarizes this discussion. We find that the theoretical time complexities of the considered frameworks are of the form of $\mathcal{O}(N_{X,P,H}^2 \cdot N_A \cdot N_{S,J,O})$. Therefore the quadratical terms N_X^2 , N_P^2 , N_H^2 are the critical ones to be analyzed. In large environments, such as the one of $10.000m^2$ used as a test bench of this paper, a complete discretization, with discretization steps of $\Delta xy = 0.5m$, and $\Delta\theta = 5^\circ$, would result in $N_X \sim 3 \cdot 10^6$ states. In our implementation, the particle filter localization needs about $N_P \sim 5000$ particles. Several executions of the particle filter with the clustering step have resulted in a number of hypotheses of about $N_H \sim 20$ in the testbench environment, thus $N_H \ll N_P \ll N_X$, indicating that the presented approach entails a significant improvement in time complexity, a key requirement in large environments.

TABLE I
COMPARISON BETWEEN EXISTING ACTIVE METHODS

	Theoretical Time Complexity	Practical Time Complexity
[12]	$\mathcal{O}(N_X^2 \cdot N_A \cdot N_S)$	$\mathcal{O}(N_X \cdot N_{Xg} \cdot N_A)$
[13]	$\mathcal{O}(N_P^2 \cdot N_A \cdot N_J)$	$\mathcal{O}(N_P^2 \cdot N_A)$
[proposed]	$\mathcal{O}(N_H^2 \cdot N_A \cdot N_O)$	$\mathcal{O}(N_H^2 \cdot N_A \cdot (N_{LS} + N_C))$

VI. SIMULATION RESULTS

Fig. 4 shows an execution of the particle filter and the clustering steps. In this situation, the simulated robot is at position $(46, 19, 90^\circ)^m$. The robot has generated 14 position hypotheses, using only its onboard sensors. These hypotheses are located along the five corridors of the environment that have very similar appearance from a laser scanner point of view. Thanks to the compass device, all hypotheses are relatively well aligned with the robot frame. This figure also labels the map and the robot frames and details the position of the cameras $C_1..C_5$.

With the situation of the figure 4, the active strategy has been evaluated for two cases, when the cameras are switched off, the *single robot* case, and when the cameras are switched on, the *cooperative case*, simulating a camera network or other well positioned robots. For both cases, the active strategy has been computed ten times in order to validate its

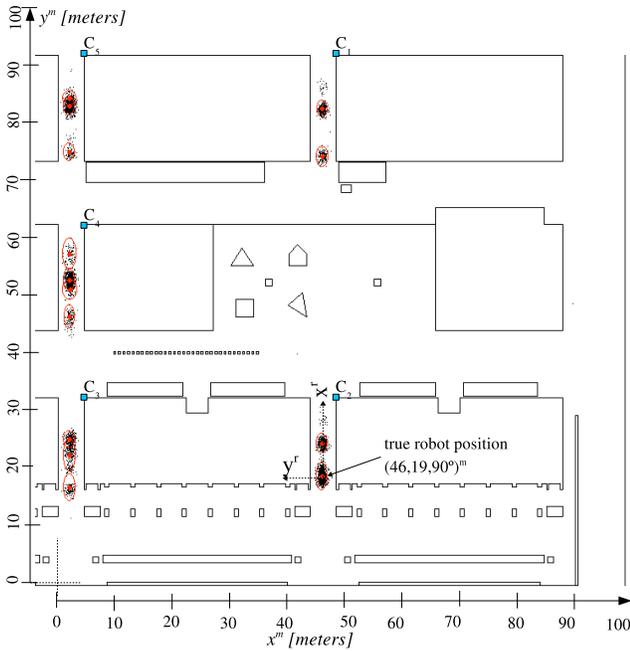


Fig. 4. Hypotheses generation step resulting in $N_H = 14$.

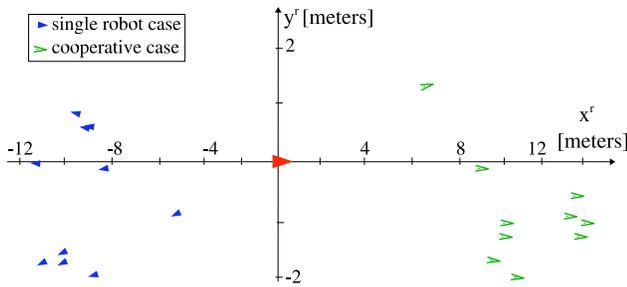


Fig. 5. The selected action to solve the situation of the figure 4. Ten executions for each case, the single robot and the cooperative one.

behaviour and stability. Fig. 5 shows the selected action for each execution in the robot frame, for both, the single robot case and the cooperative one. In the single robot case, the best actions are related on going to distinctive places of the environment in order to disambiguate the hypotheses set, so the actions drive the robot down of the corridors. Otherwise, in the cooperative case, actions are more related on going to places where a detection of the robot by the camera network is expected, thus the best actions are go up the corridors.

VII. CONCLUSIONS

This paper presents a probabilistic and active approach to the map-based global localization problem for mobile robots operating in large environments. The formulation is general, not depending on the sensory system neither on the environment model, and is extended to the cooperative case.

The first contribution of the present work is related to the reduction of the computational complexity, both in time and in memory. The improvement in time complexity is because of the active strategy reasons in terms of position

hypotheses, instead of using the whole belief, and performs computations in the observation space, which remains fixed with increments of the environment size. In terms of memory complexity, the proposed implementation uses a GIS-based world representation which is extremely compact.

The second contribution refers to the extension of the action selection approach to the cooperative case, nowadays a challenging research field for robotic community. The proposed action selection approach shows a cooperative behaviour when a lost robot operates in an environment where other robots or a sensor network are deployed, with no significant overhead in computational complexity.

REFERENCES

- [1] D. Filliat and J.-A. Meyer, "Map-based Navigation in Mobile Robots. I. A review of localization strategies," *Cognitive Systems Research*, vol. 4, December 2003.
- [2] J. Levinson, M. Montemerlo, and S. Thrun, "Map-Based Precision Vehicle Localization in Urban Environments," in *Proceedings of the Robotics: Science and Systems Conference*, (Atlanta, USA, June, 2007.).
- [3] J. Yun and J. Miura, "Multi-Hypothesis Outdoor Localization using Multiple Visual Features with a Rough Map," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (Roma, Italy, April, 2007.).
- [4] P. Jensfelt and S. Kristensen, "Active Global Localisation for a Mobile Robot Using Multiple Hypothesis Tracking," *IEEE Transactions on Robotics and Automation*, vol. 17, October 2001.
- [5] K. Arras, J. Castellanos, M. Schilt, and R. Siegwart, "Feature-based multi-hypothesis localization and tracking using geometric constraints," *Journal of Robotics and Autonomous Systems*, vol. 44, pp. 41–53, 2003.
- [6] D. Fox, W. Burgard, and S. Thrun, "Markov Localization for Mobile Robots in Dynamic Environments," *Journal of Artificial Intelligence Research*, vol. 11, pp. 391–427, 1999.
- [7] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, pp. 99–141, 2001.
- [8] E. Menegatti, M. Zoccarato, E. Pagello, and H. Ishiguro, "Image-based Monte Carlo localisation with omnidirectional images," *Journal of Robotics and Autonomous Systems*, no. 48, 2004.
- [9] A. Gasparri, S. Panzneri, F. Pascucci, and G. Ulivi, "A Hybrid Active Global Localisation Algorithm for Mobile Robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (Roma, Italy, April, 2007.).
- [10] M. Rao, G. Dudek, and S. Whitesides, "Randomized Algorithms for Minimum Distance Localization," in *Proceedings of the International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, (Nagoya, Japan, October, 2003.).
- [11] J. O'Kane, "Global localization using odometry," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (Pasadena, USA, May, 2008.).
- [12] D. Fox, W. Burgard, and S. Thrun, "Active Markov Localization for Mobile Robots," *Robotics and Autonomous Systems*, vol. 25, no. 3-4, pp. 195–207, 1998.
- [13] J. Porta, J. Verbeek, and B. Kröse, "Active Appearance-Based Robot Localization Using Stereo Vision," *Autonomous Robots*, vol. 18, no. 1, pp. 59–80, 2005.
- [14] A. Sanfeliu and J. Andrade-Cetto, "Ubiquitous networking robotics in urban settings," in *Proceedings of the IEEE/RSJ IROS Workshop on Network Robot Systems*, (Beijing, China, October, 2006.).
- [15] A. Corominas Murtra, J. Mirats Tur, and A. Sanfeliu, "Action Evaluation for Mobile Robot Global Localization in Cooperative Environments," *Journal of Robotics and Autonomous Systems, Special Issue on Network Robot Systems (accepted)*, 2008.
- [16] J. Maantay and J. Ziegler, *GIS for the urban environment*. Environmental Systems Research Institute, 2005.
- [17] B. Gärtner, "Fast and robust smallest enclosing balls," in *Proceedings of the European Symposium on Algorithms (ESA)*, (Prague, Czech Republic, July, 1999.).
- [18] S. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.