Action Evaluation for Mobile Robot Global Localization in Cooperative Environments

Andreu Corominas Murtra[1], Josep M. Mirats Tur[1], Alberto Sanfeliu[1,2]

[1] *Institut de Robòtica i Informàtica Industrial, IRI (UPC-CSIC). C/Llorens i Artigas, 4-6. 08028. Barcelona, Spain. www-iri.upc.es*
[2] *Universitat Politècnica de Catalunya, UPC. Departament d'ESAII. C/Pau Gargallo, 5. 08028. Barcelona, Spain. www.upc.edu*

Abstract

This work is about solving the global localization issue for mobile robots operating in large and cooperative environments. It tackles the problem of estimating the pose of a robot or team of robots in the map reference frame, given the map, the real-time data from the robot onboard sensors and the real-time data coming from other robots or sensors in the environment. After a first step of position hypotheses generation, an efficient probabilistic active strategy selects an action, for a single lost robot case, or two joint actions when two lost robots are in a line of sight, so that the hypotheses set is best disambiguated. The action set is adapted to the multi-hypothesis situation and the action evaluation takes into account the remote observations available in robot network systems. The paper presents the theoretical formulation for both, the non cooperative and cooperative cases. An implementation of the proposed strategy is discussed and simulation results presented.

Key words: global localization, active, cooperative, network robot system

1 INTRODUCTION

The ability of a mobile robot to solve a map-based navigation issue involves three main tasks: global localization, path planning and path execution, the

[1] This research was conducted at the Institut de Robòtica i Informàtica Industrial of the Technical University of Catalonia (UPC) and Consejo Superior de Investigaciones Científicas (CSIC). It was partially supported by Consolider Ingenio 2010, project CSD2007-00018, CICYT project DPI2007-614452, and IST-045062 of the European Community Union

later requiring two sub-tasks, position tracking and obstacle avoidance. This paper adresses the map-based global localization challenge, identified as a key topic by the robotics research community, especially in large areas [4]. Specifically, the paper studies the global localization task for mobile robots operating in large and cooperative environments. This refers to the problem of estimating the position of a robot in the map reference frame, $(x_r^m, y_r^m, \theta_r^m)$, given the map, the real-time data from the robot onboard sensors and the real-time data coming from other robots or sensors in the environment. Moreover, providing autonomy to mobile robots implies that they should solve the global localization task without human aid, therefore the paper proposes an active strategy that selects actions to be performed by the robots to solve the global localization task, taking advantage of the cooperative environment.

Different reasons have made to formulate global localization in a stochastic context: noisy sensor data, delays in the data acquisition processes or approximations of the robot kinematic model. Due to these reasons, global localization methods have to deal with uncertainty in the state space of the robot. This uncertainty is usually represented whether with a multi-gaussian distribution [10,1,26], a complete discretization of the state space [7] or by means of a set of weighted particles sampling the state space [25,15,12]. Whichever the case, the distribution is considered multi-peak for robust solutions, that is, we have to deal with multiple position hypothesis in order to solve the global localization issue. Therefore, if we are considering an autonomous robot, an active strategy is required so as to decide which command should the robot perform, in robot coordinates frame since the robot is not yet localized, to drive it to other places of the environment with the aim of disambiguating the multi-hypothesis situation. Figure 1 shows an usual multi-hypothesis situation when particle filter localization is executed in an environment of $10.000m^2$.

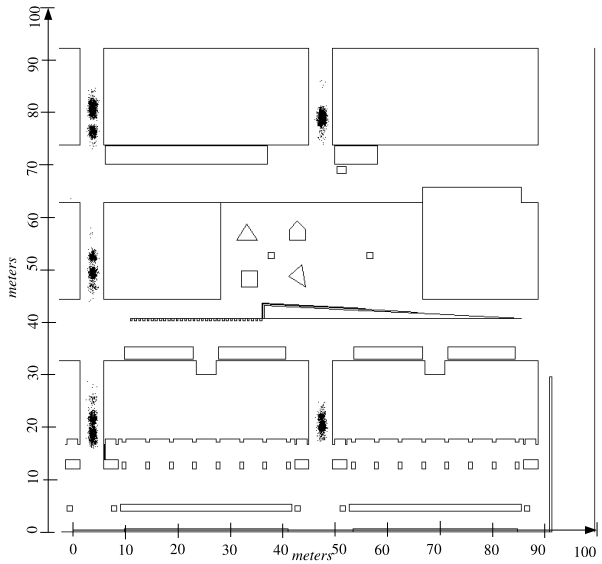

Fig. 1. Multi-hypothesis situation in global localization with particle filtering

Existing methods for active global localization can be divided in heuristic, geometric and entropy-based approaches. *Heuristic* strategies have been proposed by [10,9]. The former describes a set of rules that command the robot to the non visited areas where more features are expected to be found, while the later drives the robot to the nearest obstacle found in the map, taking into 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 [20,17]. In [20], 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, [17] 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 present simulated experiments in an ideal polygonal world. Extrapolation of these approaches to uncertain, dynamic and cooperative environments remains an open question.

A third family of active techniques, called *entropy-based*, address the issue of the uncertainty. In [6], a general formulation of the problem is presented, based on the Markov framework of the same authors [7]. They evaluate a finite and fixed set of actions computing the expected future *belief* and its associated entropy. The strategy selects the action minimizing the expected entropy, so selecting the action that most concentrate the expected position distribution over the state space. The complete discretization of the state space provokes high computational complexity, especially for large environments. Authors in [18] implement an entropy-based active localization algorithm 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. These approaches grow in complexity when multiple robots are operating in the same environment and detections should be considered for the computation of the expected belief.

In recent years, attention has been paid to cooperative robotics. A group of robots is considered to solve a common task or to provide services to a user community. Moreover, when a set of sensors is deployed on the environment, robots and sensors build a computer network, and a *network robot system* arises. In such a system, cooperative behaviours can appear in two ways: *sharing information* among the sensors and robots of the network or *selecting joint actions* to solve a common task. Cooperative localization has been adressed by means of different passive approaches. Some authors have studied the relative localization and identification between robots [19,21,22], a key subsystem to integrate observations coming from a remote observer (another robot or the

sensor network). In [5] map-based global localization is addressed, where a lost robot integrates observations from another robot using a particle filter framework. A simulated experiment shows how a group of heterogeneous robots can localize themselves, thanks to the proposed cooperative framework, in an environment where, otherwise, they could not localize. Other authors also adressing the information-sharing issue for cooperative localization are [14,23]. They propose fusion frameworks such as decentralized Kalman Filters focused on applications in open outdoor terrain, using absolute sensors, such as GPS or compass, to correct estimations. Finally, the RoboCup community has also extensively studied the multi-robot localization problem but in a more limited way, because of the particular specifications of such environment [3].

Up to the knowledge of the authors, there are no works adressing the *active* global localization in such a netwok robot system, neither for a single lost robot member of that network, where the robot can take benefit of the potentialities of remote observations in the action selection step, nor for the case of more than one lost robot, members of that network, where the robots furthermore can select a set of joint actions, one action to be execute by each lost robot.

The present work proposes a *novel* and *efficient* solution to the active map-based global localization for cooperative mobile robots operating in large environments. Since the goal application is to navigate in large environments, such as urban settings, with a group of robots operating on it and/or with a deployed sensor network [24], both computational efficiency and cooperation are addressed. Novelty relies on the consideration of a cooperative environment in the action selection step for global localization, in the two described ways of cooperation: sharing information and selecting joint actions. Efficiency is thanks to the reduction of time complexity for the active strategy, since computing is performed in terms of the number of position hypotheses, instead of using the whole belief function such as entropy-based approaches do.

The paper is organized as follows: In section 2, the basic assumptions, main definitions and notation used are presented for the proposed active and cooperative global localization method. Section 3 presents the theoretical framework for the single robot case in a non cooperative environment and section 4 extends the formalism to the network robot context (cooperative environment), developing the single lost robot case and the two lost robots case. An implementation of the presented active strategy is proposed in section 5. Section 6 describes the simulator used to test the proposed approach in a network robot environment. Section 7 analyses the computational complexity of the presented approach for both, the theoretical framework and the proposed implementation. Simulation results in the URUS [24] test bench area, an outdoor campus environment of 10.000 m^2, are presented in section 8 showing promising behaviours. Finally, section 9 lists the main conclusions pointing out the potentialities of the presented work.

2 BASIC ASSUMPTIONS AND DEFINITIONS

We give in this section the basic considered assumptions as well as the formal definitions of the concepts that will be used all throughout the paper dealing with the presented active strategy. Some of the assumptions made are common to other existing map-based mobile robot localization approaches. Please note that the points listed below are further discussed in section 5.

2.1 Basic assumptions

- Environment is large-scale since it goes beyond the robot sensor horizons.
- The robot has a *map*, $\mathcal{M}$, as a data base describing the area where it operates following an environment model. There is no restriction on the ontology of this environment model.
- The robot is equipped with a sensory system that provides proprioceptive observations at time t, in order to propagate the state of the robot using a kinematic model of the robot platform.
- The sensor network does not cover all the working area. Otherwise, the robots are considered to have always communication coverage, so data communication service is always available.
- Position of the sensor network devices is known with a given uncertainty.
- A sub-system for robot identification and relative localization is assumed to be onboard the robots and also implemented by the sensor network. If a robot is within an area covered by the sensor network or in the line of sight of another robot, the first one can request for remote observations about its position. Examples of such a sub-system can be found in [19,21,22].
- A robot can process *real* observations coming either from its onboard exteroceptive sensor readings, from other robots, or from the sensor network server, thanks to an implemented data communication service. Figure 2 illustrates this issue, remarking the difference between local/remote sensor readings of the whole system and local/remote observations available at a given robot for local processing.

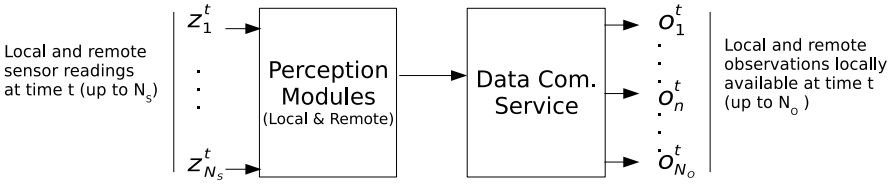

Fig. 2. Locally available observations for a mobile robot in a network robot context

- An explicit model for each observation is assumed, in order to compute *synthetic* observations, provided that the robot is at a given state. These observation models can be pre-computed off-line and stored in a look-up table or in an appearance map, or they can be computed at execution time.

While the former alternative needs discretization of the full state space to compute the model, appearing a trade-off, especially in large enviroments, between discretization resolution and memory complexity, the later approach increases time complexity.

- Studying the errors provoked by data communication delays is beyond the scope of this work. Here, the robots are assumed to move relatively slow in comparison with communication delays, thus errors on integrating delayed data are directly modelled incrementing uncertainty of observations.
- A path planning technique is assumed to be available in order to provide the robot a collision free path to execute.

2.2 Definitions

- A position p in the map frame coordinates is $X_p^m = (x_p^m, y_p^m, \theta_p^m)$. The *true* state of the r^{th} robot is defined by its position $X_r^m = (x_r^m, y_r^m, \theta_r^m)$. The *estimated* state of the r^{th} robot is $\hat{X}_r^m = (\hat{x}_r^m, \hat{y}_r^m, \hat{\theta}_r^m)$.
- The state space is given by $X^m = \{[x_{min}^m, x_{max}^m], [y_{min}^m, y_{max}^m], (-\pi, \pi]\}$. Both X_r^m and $\hat{X}_r^m$ are in X^m.
- A proprioceptive observation at time t is denoted by o_0^t and is the input to propagate the state of the robot using a kinematic model $f(\cdot)$.
- At time t, a robot can use up to N_O *real* observations, $o_n^t, n = 1..N_O$. Each observation is in its observation space, $o_n^t \in O_n$.
- The n^{th} *synthetic* observation at the state X_p^m, is computed by the explicit observation model and is denoted as $o_n^s(X_p^m)$, where $o_n^s(X_p^m) \in O_n$.
- A likelihood function $L_n(\cdot)$ is defined that calculates the matching between two n^{th} observations (real and/or synthetic). This function approaches to 1 for similar observations and goes close to 0 for distinctive ones.

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

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

$$p(o_n^t|X_p^m) = L_n(o_n^t, o_n^s(X_p^m)) \ \in \ [0, 1] \tag{2}$$

- 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:

$$p(o_n^s(X_q^m)|X_p^m) = L_n(o_n^s(X_q^m), o_n^s(X_p^m)) \ \in \ [0, 1] \tag{3}$$

- A set of N_H position hypotheses for the robot is defined as $H = \{h_1, \ldots, h_{N_H}\}$ where the i^{th} hypothesis is defined with a position in the map coordinate

frame, $X^m_{h_i} = (x^m_{h_i}, y^m_{h_i}, \theta^m_{h_i})$, a covariance matrix, C_{h_i}, and a probability associated to that hypothesis such that for the robot position, p_{h_i}:

$$h_i = \{X^m_{h_i}, C_{h_i}, p_{h_i}\},\ \forall i = 1..N_H;\ \ \sum_{i=1}^{N_H} p_{h_i} = 1 \tag{4}$$

Different approaches can be found in the literature providing this hypothesis set explicitly [10,1,26], or, alternatively, clustering a particle set such as the provided by the particle filter localization methods [25].

With all these assumptions and definitions, the problem to be solved by an active strategy is where to move a lost robot in order to reduce the hypotheses set. The proposed strategy exploits the map and the cooperative environment, selecting actions that drive the robot where *distinctive* observations are expected among the hypotheses. The proposed active approach is formulated in a general way but section 5 discusses practical issues when implementing the above listed requeriments in order to obtain the illustrative results of section 8.

3 ACTIVE STRATEGY. NON COOPERATIVE ENVIRONMENT

This section formulates the active strategy for the single robot case operating in a non cooperative environment, therefore only observations coming from its own sensors are available. The proposed active strategy is divided in three steps, and only one action can be selected. The first step consists in randomly generating a set of *exploration particles* in the robot coordinate 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 coordinate frame, with minimum expected number of remaining hypotheses is the selected one to drive the robot.

3.1 Generating Exploration Particles

Let's call the k^{th} *exploration particle*, ϵ^r_k, as a random position in the robot coordinates frame generated within a given disk of radius R_ϵ around the robot. R_ϵ is called the *exploration radius*.

$$\epsilon^r_k = X^r_{\epsilon_k} = (x^r_{\epsilon_k}, y^r_{\epsilon_k}, \theta^r_{\epsilon_k}) \tag{5}$$

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

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

Please note that $\epsilon_{ki}^m \in X^m$ and 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 coordinates frame, since we have to consider all hypotheses and, therefore, we should translate ϵ_k^r for each hypotheses $h_i, i = 1..N_H$. Fig. 3 illustrates this observation.

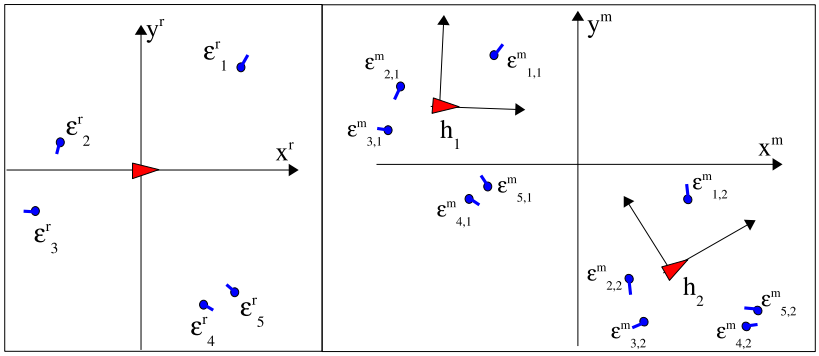

Fig. 3. A set of 5 exploration particles in the robot coordinates frame (left) and their transformation to map coordinates frame (right) when $N_H = 2$. Each exploration particle is represented as a point with a short line indicating its orientation.

3.2 Multi-hypothesis Path Planning

Even if ϵ_k^r is expressed in the robot coordinate frame and, therefore, the robot knows where the exploration particle is positioned, since ϵ_k^r can be beyond the sensor horizons we have to assure that a free path exists between the robot and ϵ_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 coordinate frame using all hypotheses constraints. Figure 4 draws the MHPP approach in an illustrative geometric world. For this step, C_{h_i} can be used as a clearance factor. If a multi-hypothesis path (MHP) exists between the robot and the ϵ_k^r, we label ϵ_k^r as a valid candidate destination, e_k^r, to drive the robot adding it to the set of all valid exploration particles E. Summarizing, the output of the first and second steps of the active strategy will be a set E of N_E exploration particles $E = \{e_1^r \dots e_{N_E}^r\}$ that are connected to the robot with a MHP. This set E is the action set to be evaluated, since each e_k^r is considered as an action *go to* e_k^r. This action set has been automatically generated and it is adapted to

the current situation of the robot. Note that E is not a fixed action set such as most of the previous works proposed in the literature. Please, remind that an exploration particle is expressed in the robot frame and it can be translated to the map frame if we assume hypothesis h_i being true:

$$e^m_{ki} = e^r_k|h_i,\ e^m_{ki} \in X^m; \quad \forall k = 1..N_E, \forall i = 1..N_H \tag{7}$$

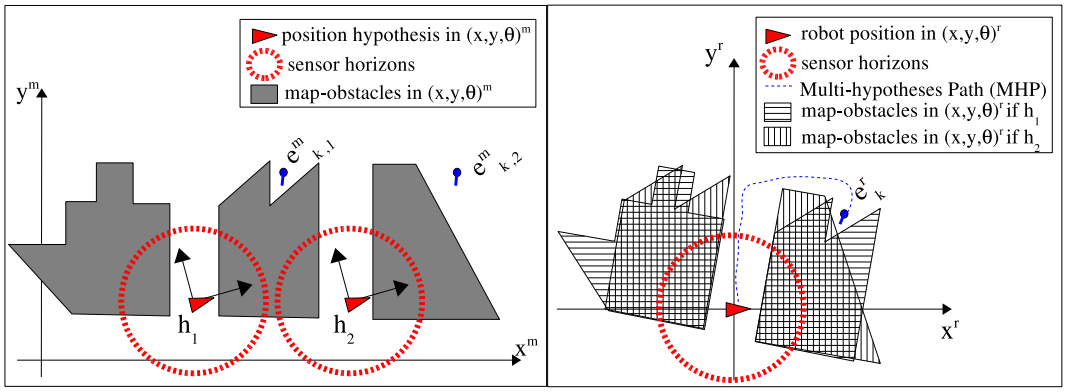

Fig. 4. Multi-hypotyhesis Path (MHP) in an illustrative geometric world. Map coordinate frame on the left and robot coordinate frame on the right.

3.3 Computing Hypotheses Reduction

The goal of this third step is to compute $\hat{N}_H(e^r_k)$, as the expected number of remaining hypotheses given that the robot goes to e^r_k and senses the environment. To compute this number, we first must to be able to compute $\hat{N}_H(e^r_k|h_i)$, as the expected number of remaining hypotheses assuming h_i as the true position hypothesis, and given that the robot will execute the action *go to* e^r_k. Using equation 3 and considering we are integrating only one exteroceptive observation ($N_O = 1$), we formulate:

$$\hat{N}_H(e^r_k|h_i) = \sum_{j=1}^{N_H} p(o^s_1(e^m_{kj})|e^m_{ki}) \tag{8}$$

If the perception module of the robot provides N_O exteroceptive observations, and we assume independency between them, equation 8 is generalized as:

$$\hat{N}_H(e^r_k|h_i) = \sum_{j=1}^{N_H} \prod_{n=1}^{N_O} p(o^s_n(e^m_{kj})|e^m_{ki}) \tag{9}$$

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}:

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

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 2. 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 $\hat{N}_H(e_k^r)$.

4 ACTIVE STRATEGY. COOPERATIVE ENVIRONMENT

This section formulates the previous strategy for a cooperative context in which different robots work in a network robot environment. A network robot environment is formed by a sensor network of N_C sensors and a group of N_R robots. The formulation is presented for the two ways of cooperation discussed in section 1: sharing information and selecting joint actions.

4.1 Single Lost Robot in a Sensor Network: Sharing Information

This subsection analyses the particular case of a lost robot which is a member of a network robot system. In this situation the active strategy selects, as in section 3, one action exploiting its onboard sensors and the map, but also uses the potentialities of integrating remote observations. Let's define the coverage space of the sensor network, which does not depend on time, as:

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

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

$$C^t_{RN} = \bigcup_{r=1}^{N_R} C^t_r \ , \quad C^t_{RN} \subset X^m \tag{12}$$

where C^t_r is the coverage area of the r^{th} robot at time t. For a lost robot, $C^t_r = \emptyset$.

In the proposed network robot system, both C_{CN} and C^t_{RN} are data available on the central server, since it knows where the sensors are deployed and where the non lost robots are. Note that a robot can request both coverage spaces at a given time and, therefore, 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 section 3. 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^m_{kj} is in $C_{CN} \bigcup C^t_{RN}$. If this is the case, a remote observation for that position is available and the $p(o^s_n(e^m_{kj})|e^m_{ki})$ can be computed where o^s_n is the model for that remote observation.

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

The reader can consider, as an illustrative example, the GPS system as 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 out of coverage will be attracted by actions driving the robot to areas where GPS is available.

4.2 Two Lost Robots: Selecting Joint Actions

This subsection addresses the case where two lost robots are performing the global localization task, hence trying to locate themselves in the environment.

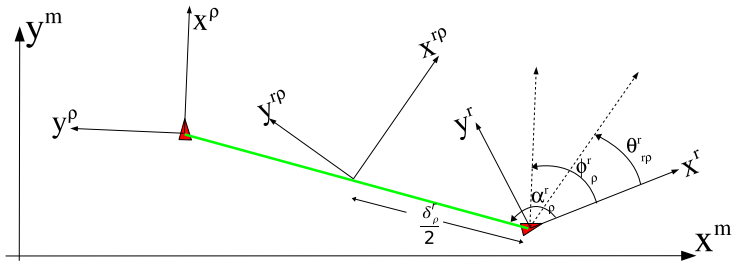

Fig. 5. Robot frames and the cooperative robot frame.

In this case the output of the active strategy should be two joint actions, each one to be executed by each robot.

We assume two lost robots, r^{th} and ρ^{th}, that are in the line of sight of each other and can be detected between them by means of an active or passive beacon system like the ones reported in [19,21,22]. So they can build up a commom frame in the middle point of this line of sight, called the cooperative robot frame and denoted by $(x^{r\rho}, y^{r\rho}, \theta^{r\rho})$(see figure 5). From the observation made by the r^{th} robot to the ρ^{th} robot, $(\delta^r_\rho, \alpha^r_\rho, \phi^r_\rho)$, the cooperative frame is placed in terms of the r^{th} robot coordinates as:

$$x^r_{r\rho} = \frac{\delta^r_\rho}{2} cos\alpha^r_\rho; \quad y^r_{r\rho} = \frac{\delta^r_\rho}{2} sin\alpha^r_\rho; \quad \theta^r_{r\rho} = \frac{\phi^r_\rho}{2} \tag{13}$$

Initially, the robots translate their own observations to the cooperative frame $(x^{r\rho}, y^{r\rho}, \theta^{r\rho})$ and an hypotheses generation step to localize its center is executed integrating all available observations. Once a set of hypotheses for $(x^{r\rho}, y^{r\rho}, \theta^{r\rho})$ is obtained, the active strategy could run as exposed previously and the robots could perform the two best actions, a different action for each robot. But this approach is not suitable since this couple of actions could disambiguate the same subset of hypotheses and, therefore, the two decided actions would be redundant. Instead, it would be desirable that the robots take complementary actions, complementary in the sense that each action disambiguates a different subset of hypotheses. Previous strategy of estimating $\hat{N}_H(e^r_k)$ gave us a *quantitative* criterion but for the multi-action problem we need also a *qualitative* criterion to evaluate the actions.

To formulate a qualitative criterion, we start defining a N_H-dimensional vector, $V(e^{r\rho}_k)$, for each exploration particle in the cooperative robot frame. The i^{th} component of such vector is defined using the equation 9 as:

$$V(e^{r\rho}_k)_i = \hat{N}_H(e^{r\rho}_k|h_i) - 1, \ \forall i = 1 \dots N_H, \ \forall k = 1 \dots N_E \tag{14}$$

This vector indicates at its i^{th} component how the k^{th} exploration particle resolves the hypothesis h_i. Substraction of 1 removes the contribution of $i = j$ of equation 9, which is always 1, and leads to a more sparse vector set, more suitable for computations presented below. A vector defined by the above

equation will have the following properties inherited from equations 1 and 3:

- $V(e_k^{r\rho})_i \in [0, N_H - 1]\ ,\ \forall i = 1..N_H,\ \forall k = 1..N_E$
- The ideal exploration particle is that $e_k^{r\rho}$ which fully disambiguates the situation. It has a vector $V(e_k^{r\rho}) = [0]$.
- An useless exploration particle has completely ambiguous synthetic observations. It has a vector $V(e_k^{r\rho}) = [N_H - 1]$.

We define a weighted scalar product between two of these vectors as:

$$< V(e_k^{r\rho}), V(e_q^{r\rho}) >= \sum_{i=1}^{N_H} V(e_k^{r\rho})_i \cdot V(e_q^{r\rho})_i \cdot p_{h_i}, \quad \forall k, q = 1 \ldots N_E \tag{15}$$

These weighted scalar product gives us a measure of the complementariness of these two exploration particles. The scalar product will be maximized when two particles disambiguate the same subset of hypotheses and, therefore, their vectors will be colinear. Otherwise, when two exploration particles disambiguate different subsets of hypotheses, $V(e_k^{r\rho})$ and $V(e_q^{r\rho})$ are close to be orthogonal and the scalar product approaches to zero.

For this case, the strategy initially selects the best exploration particle in the sense of minimum $\hat{N}_H(e_k^{r\rho})$, as it was done for the single robot case, which will be called the primary one, $e_{s1}^{r\rho}$. Then, the strategy will search a support action, that is, going to the exploration particle $e_k^{r\rho}$ minimizing the weighted scalar product with $e_{s1}^{r\rho}$. This support exploration particle will be labelled as $e_{s2}^{r\rho}$. Finally, $e_{s1}^{r\rho}$ and $e_{s2}^{r\rho}$ have to be executed by the robots so it is necessary to compute the translation of both actions from the cooperative robot frame $(x, y, \theta)^{r\rho}$ to each robot frame $(x, y, \theta)^r$ and $(x, y, \theta)^\rho$.

5 IMPLEMENTATION OF THE ACTIVE STRATEGY

This section details the implementation of the proposed active strategy, which is intended to be independent from the origin of data, whether coming from a real platform and sensors or from a simulated environment.

5.1 Environment Model

The experimentation area is the surroundings of the 'FIB square' at the Campus Nord of the Universitat Politècnica de Catalunya. This outdoor environment (see Fig.1) is about $100m$x$100m$ and it is the test bench environment for the URUS european project [24]. The environment is modelled with the vector format of the Geographical Information Systems (GIS) model [13]. One

of the benefits of using such a map is that the robots are using a standard format, human compatible, and usually available at city councils of important towns, instead of an adhoc representation adapted to the robot sensors. Other interests of this representation is its compactness ($\sim 4bits/m^2$ in our case), required for large environments, and the possibility to perform both localization and path planning tasks on it. In such map, the environment is represented by a set of obstacles. Each obstacle is represented with some related semantic information and a set of straight segments limiting its borders. Each segment is parametrized with two points in the map coordinate frame, height information, and is also accompanied of some semantic data. In order to drastically reduce computational cost of routines dealing with the map, each obstacle is enclosed in a minimum bounding ball [8]. A detailed exposition of this representation can be found in [2].

5.2 Observation Models

For the active strategy we take into account a laser scanner, an electronic compass and a set of omnidirectional cameras deployed on the environment. The relative localization between robots is only considered to build the cooperative frame for the two lost robots case. This subsection describes the observation models used to compute the synthetic observations $o_n^s(X_p^m)$, that is the n^{th} observation expected to have at the state X_p^m. These models are used to compute the $p(o_n^s(e_{kj}^m)|e_{ki}^m)$ of equation 9.

The observation model for the **laser scanner**, $o_1^s(X_p^m)$, is a vector of $N_{LS} = 133$ ranges over the scan aperture of $(-95, 95)$ degrees. The maximun laser range is limited to $R_L = 15m$. To compute this observation model, we compute the ray tracing function from the position X_p^m. Ray tracing uses the map to return the distance between the given position and the nearest obstacle forward. In order to obtain an entire scan observation, the ray tracing is performed N_{LS} times covering all the scan aperture. The output of the observation model of the laser scanner is, for the X_p^m position:

$$o_1^s(X_p^m) = (o_{1,1}^s(X_p^m)...o_{1,N_{LS}}^s(X_p^m));\ o_{1,i}^s(X_p^m) = rayTracing(X_p^m, \mathcal{M}, i) \quad (16)$$

For the **compass**, the observation model, $o_2^s(X_p^m)$, is:

$$o_2^s(X_p^m) = \theta_p^m, \quad \in (-\pi, \pi] \quad (17)$$

The **sensor network** is modelled as a set of N_C omnidirectional cameras deployed at known positions of the environment. The implemented observation model, $o_3^s(X_p^m)$, outputs a triplet containing range, bearing and heading

measures (see figure 6). The coverage area for the c^{th} camera, C_c, is modelled as the set of positions of the state space that are in a line of sight of length less than $R_C = 7m$ with the position of the c^{th} camera. If $X_p^m \in C_c$, the observation model is computed as:

$$o_3^s(X_p^m) = \begin{bmatrix} \delta_{c,p} \\ \alpha_{c,p} \\ \phi_{c,p} \end{bmatrix} = \begin{bmatrix} \sqrt{(x_p^m - x_c^m)^2 + (y_p^m - y_c^m)^2} \\ atan(\frac{y_p^m - y_c^m}{x_p^m - x_c^m}) \\ \theta_p^m \end{bmatrix} \quad (18)$$

otherwise, when $X_p^m \notin C_c$, $\forall c = 1..N_C$ the output of the model is $o_3^s(X_p^m) = (-1, -1, -1)$, indicating that the position is not seen by the camera network.

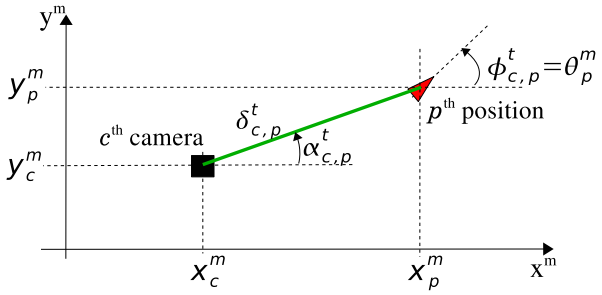

Fig. 6. The observation model for the c^{th} camera seeing the position X_p^m.

This implementation does not integrate potential **relative localization** between robots in equation 9 as observations. This could be done in a similar way such that implemented by the cameras since a well localized robot can be considered as an static camera of the camera network.

5.3 Likelihood Functions

In order to compute the conditional probabilities of equation 9, and as stated in equations 2 and 3, we need to implement likelihood functions $L_n(\cdot)$ for each of the above mentioned observations models. All the implemented $L_n(\cdot)$ are based on the well known $erfc()$. Below, these likelihoods are presented as a functions between a real and a synthetic observation but, as discussed in equation 3, they can be also computed for two synthetic observations.

The likelihood function for the laser observations, where $\sigma_1 = 0.5m$, is:

$$L_1(o_1^t, o_1^s(X_p^m)) = \frac{1}{N_L} \sum_{i=1}^{N_{LS}} erfc(\frac{|o_{1,i}^t - o_{1,i}^s(X_p^m)|}{\sigma_1 \sqrt{2}}) \quad (19)$$

For the compass observations, the $L_2(\cdot)$, where $\sigma_2 = 0.08rad$, is:

$$L_2(o_2^t, o_2^s(X_p^m)) = erfc(\frac{|o_2^t - o_2^s(X_p^m)|}{\sigma_2\sqrt{2}}) \tag{20}$$

The likelihood function for the camera network observation, where $\sigma_{3,1} = 0.5m$, $\sigma_{3,2} = 0.05rad$ and $\sigma_{3,3} = 0.1rad$, is:

$$L_3(o_3^t, o_3^s(X_p^m)) = \frac{1}{3}\sum_{i=1}^{3} erfc(\frac{|o_{3,i}^t - o_{3,i}^s(X_p^m)|}{\sigma_{3,i}\sqrt{2}}) \tag{21}$$

For the $L_n(\cdot)$ involving two synthetic observations $o_n^s(e_{ki}^m)$ and $o_n^s(e_{kj}^m)$, parameters σ_n can be calculated using the matrices C_{h_i} and C_{h_j}, $\forall i, j, k, n$.

5.4 Hypotheses Generation: Particle Filtering and Clustering

A particle filter has been implemented to perform an initial search as previous step of the active strategy, following the well stablished framework of [25]. The filter represents the position belief with a set of N_P particles, $s_i^t = \{X_{s_i}^{m,t}, w_{s_i}^t\}, \forall i = 1..N_P$. It initializes generating the set of position particles sampling randomly the $(x, y)^m$ space, but sampling the θ^m space with a normal law of $\mathcal{N}(\theta^0, \sigma_{\theta^0})$, being θ^0 the first 'magnetically clean' compass observation. Weights of each particle are initialized to $w_{s_i} = 1/N_P$. A propagation step is performed to move each particle's state, using the odometric obervation o_0^t and the kinematic model $f(\cdot)$. The above described likelihood functions integrate real observations to correct the weights of each particle position as:

$$w_{s_i}^t = w_{s_i}^{t-1} \cdot \prod_{n=1}^{N_0} p(o_n^t | X_{s_i}^{m,t}) \ , \ \forall i = 1..N_P \ , \ \forall t > 0 \tag{22}$$

The resampling step generates a new particle set sampling the old one according to the particle weights, so likely areas are successively more sampled. We have used $N_P = 5000$ for this initial step. After several iterations, particles are concentrated in differentiated clusters, so we perform a clustering step to generate a reduced hypotheses set H. Clustering is implemented using 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 $c(K_k)$ be the centroid of it. Initially, the routine creates the first cluster using the first particle $c(K_1) = X_{s_1}^m$. The rest of the particles will join to an already created cluster if $d(X_{s_i}^m, c(K_k)) < R_K$ or, otherwise, will create a new cluster. R_K is the parameter fixing clustering size, set to $R_K = 3m$. Each time that a particle joins to an already created cluster, the centroid is updated, using all the particles

in that cluster, as a weighted mean of their positions. The fact that, in a particle filter, the more likely particles are usually at the center of the clusters improves the performance of this simple method. Finally, a covariance matrix is computed with the particle subset of each cluster, and clusters become the position hypotheses, so the system is ready to perform the active strategy.

5.5 Multi-hypothesis Path Planning

The Rapidly-Exploring Random Trees (RRT) approach [11] has been implemented. In the case of the multi-hypothesis path planning, the tree is computed in the robot coordinate frame translating the map obstacles to this frame for each hypotheses in a similar way as equation 6 does. When planning paths in the robot coordinates to reach e_k^r goals, randomly points that build iteratively the tree are generated in the surroundings of the robot, instead of on the whole map in order to improve the efficiency of the RRT. Details on the implementation of the RRT's can be found in [2]. However, we have seen that generation and validation of the exploration particles can be collapsed in a single step by means of building a single RRT bounded in the exploration area. The RRT is also generated in the robot frame, taking into account the constraints of all hypotheses. The nodes of the RRT will be directly the exploration particles. This computes a single but 'big' RRT just once, avoiding the computation of a RRT to validate each exploration particle. Exploration radius R_ϵ is set to $20m$.

6 Overview of the Simulated Environment

The implemented simulator periodically updates the position of each robot $X_r^{m,t}$, $\forall r = 1..N_R$ according the velocities commanded by the user or any other process driving the robot. This simulator also has several independent processes that compute the real observations used by the localization filter. There are also independent processes for camera network observations and for relative localization between robots. This simulator uses the YARP middelware for interprocess communication [16] and is fully implemented in C++ following the object oriented programming paradigm. Figure 7 shows the diagram of the processes involved in the simulations of the global localization. With this proposed architecture, the localization module is fully reusable when observations come from real sensors, as in the case of real experimentation.

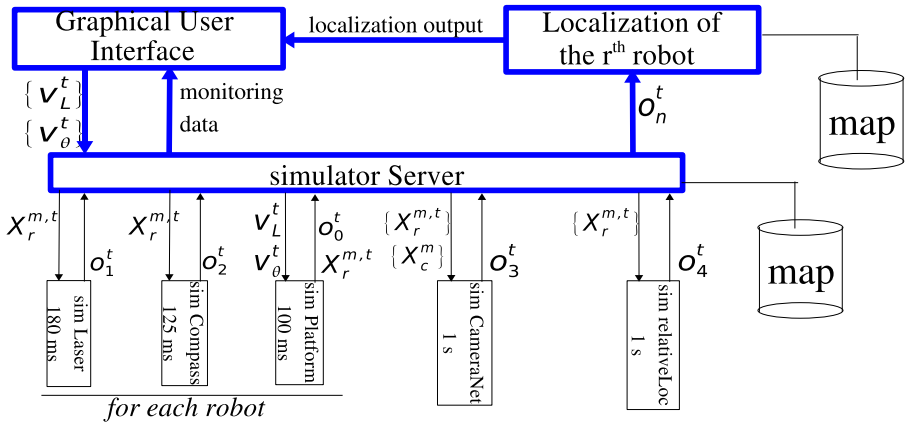

Fig. 7. Diagram of involved processes of the simulated environment. Big bold blocks are main processes and small non-bold blocks are threads. Wide arrows are YARP communications and thin arrows represent shared memory communications.

6.1 Mobile platform

We simulate a generic holonomic wheeled mobile platform, that, at time t, receive translational and rotational velocities, v_L^t and v_θ^t, to propagate, at each time step ΔT_0, its position with the kinematic model $f(X_r^{m,t-1}, v_L^t, v_\theta^t)$:

$$X_r^{m,t} = \begin{bmatrix} x_r^m \\ y_r^m \\ \theta_r^m \end{bmatrix}^t = \begin{bmatrix} x_r^m \\ y_r^m \\ \theta_r^m \end{bmatrix}^{t-1} + \begin{bmatrix} \Delta T_0 v_L^t cos(\theta_r^{m,t-1} + \Delta T_0 v_\theta^t) \\ \Delta T_0 v_L^t sin(\theta_r^{m,t-1} + \Delta T_0 v_\theta^t) \\ \Delta T_0 v_\theta^t \end{bmatrix} \tag{23}$$

When a set of N_R is being simulated, this update is done by each platform as an independent thread.

6.2 Real Observations

The onboard sensors and the other sensors and robots of the network provide real observations, o_n^t, computed from the simulated robot position, to be integrated by the localization algorithm, in the hypotheses generation step.

Each platform is equipped with **wheel encoders** that provide, at time t, the odometric observation o_0^t. This observation is composed by the increment in translational motion, δL^t, and the increment in rotational motion, $\delta\theta^t$. We add to these increments a simulated normal noise with standard deviation of 5% in translation and 10% in rotation:

$$o_0^t = (\delta L^t + \mathcal{N}(0, 0.05 * \delta L^t), \delta\theta^t + \mathcal{N}(0, 0.1 * \delta\theta^t)) \tag{24}$$

where,

$$\delta L^t = \sqrt{(x_r^{m,t} - x_r^{m,t-1})^2 + (y_r^{m,t} - y_r^{m,t-1})^2}; \quad \delta\theta^t = \theta_r^{m,t} - \theta_r^{m,t-1} \tag{25}$$

Each simulated robot has also a simulated **laser scanner** RS4 (Leuze corp.), providing, at time t, a real observation o_1^t; computed from the simulated robot position, $X_r^{m,t}$, following the model described in the section 5. However, to simulate a real observation, we add normal noise with $5cm$ standard deviation.

$$o_1^t = (o_{1,1}^t, .., o_{1,N_{LS}}^t); \quad o_{1,i}^t = rayTracing(X_r^{m,t}, \mathcal{M}, i) + \mathcal{N}(0, 0.05) \tag{26}$$

An **electronic compass** TCM2 (PNI corp.) is also simulated to be onboard of each robot and provides, at time t, a real observation o_2^t. This observation is directly computed as the heading of the simulated robot position with a normal noise of standard deviation of $0.05rad$:

$$o_2^t = \theta_r^{m,t} + \mathcal{N}(0, 0.05) \tag{27}$$

In the simulator, compass observations are always available. However, compass measurements are usually corrupted by magnetic distortions. For the local distortions, the TCM2 device provides an automatic routine to calibrate itself in a given position on the robot. For the external distortions, this device has a magnetic alarm to detect corrupted readings.

The sensor network is modelled as a set of N_C **omnidirectional cameras** deployed at known positions. This camera network provides to the robot, at time t, the observation o_3^t. The model to compute real observations of the camera network is the same than the one exposed in section 5, but adding normal noise to the measurements. If a robot is in the coverage area of the c^{th} camera, $X_r^m \in C_c$, a real observation of the camera network is available as:

$$o_3^t = \begin{bmatrix} \delta_{c,r}^t \\ \alpha_{c,r}^t \\ \phi_{c,r}^t \end{bmatrix} = \begin{bmatrix} \sqrt{(x_r^{m,t} - x_c)^2 + (y_r^{m,t} - y_c)^2} + \mathcal{N}(0, 0.5)) \\ atan(\frac{y_r^{m,t} - y_c}{x_r^{m,t} - x_c}) + \mathcal{N}(0, 0.02 * \delta_{c,r}^t) \\ \theta_r^{m,t} + \mathcal{N}(0, 0.2) \end{bmatrix} \tag{28}$$

When the ρ^{th} robot sees the r^{th} one in a line of sight of length less than $R_R = 10m$, the observation made by ρ at time t, in terms of the ρ^{th} frame, is:

$$o_4^t = \begin{bmatrix} \delta_r^{\rho,t} \\ \alpha_r^{\rho,t} \\ \phi_r^{\rho,t} \end{bmatrix} = \begin{bmatrix} \sqrt{(x_r^{m,t} - x_\rho^{m,t})^2 + (y_r^{m,t} - y_\rho^{m,t})^2} + \mathcal{N}(0, 0.05)) \\ atan(\frac{y_r^{m,t} - y_\rho^{m,t}}{x_r^{m,t} - x_\rho^{m,t}}) + \mathcal{N}(0, 0.01 * \delta_r^{\rho,t}) \\ \theta_r^{m,t} - \theta_\rho^{m,t} + \mathcal{N}(0, 0.1) \end{bmatrix} \tag{29}$$

7 COMPUTATIONAL COMPLEXITY

This section discusses the computational cost of the proposed active approach, both in time and memory, and compares it to that of the existing entropy-based methods [7,18]. In order to compare different methods, we introduce the notation N_A as the number of actions to be evaluated, which in our method coincides with N_E since each exploration particle supposes an action.

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 implementation presented in section 5 and since the observations are computed on-line, N_0 becomes $N_{LS} + N_C$, which refers to the laser scanner number of 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 compact 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 spatial representation has not been analyzed in this work but the real environment of about $10.000m^2$, used in this paper as a testbench area, is represented with a map of about $40KBytes$, supposing a very efficient map model ($4Bytes/m^2$).

For the work in [7], based on the Markov framework, 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 strategy in the sense of creating a set of Gaussians instead of having a complete 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 and this approach remains too expensive in large areas.

Using the same entropy-based approach but based on the particle representation of the uncertainty, the work presented on [18] has a time complexity of $\mathcal{O}(N_P^2 \cdot N_A \cdot N_J)$, where N_P is the number of particles representing the belief, 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 incrementing 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 [7]. However, the complexity remains high, specially for large environments where

the amount of particles needed to global localize the robot is also large. In the practical experimentation, authors in [18] report the requirement to reduce the action set and the size of the environment.

Table 1 summarizes this discussion. We find that the theoretical time complexities of the considered frameworks are of the form of $\mathcal{O}(N^2_{X,P,H} \cdot N_A \cdot N_{S,J,O})$. Therefore the quadratical terms $N^2_X,\ N^2_P,\ N^2_H$ 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 hypotyheses 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 requeriment in large environments. For the practical implementation we found a trade-off between the pre-computation of the observation models, which increases the memory complexity, and the on-line computation of these models, which increases the time complexity. In our proposed implementation we have choosen to compute on-line the observation models since the memory resources required to store these models grows up with the number of states N_X, which increases with the environmnt size, therefore being critical in large evironments.

Table 1
Comparison of computational complexities between existing active methods

	Theoretical Time Complexity	Practical Time Complexity	Practical Memory Complexity
[6]	$\mathcal{O}(N^2_X \cdot N_A \cdot N_S)$	$\mathcal{O}(N_X \cdot N_{Xg} \cdot N_A)$	$\mathcal{O}(N_X \cdot N_S)$
[18]	$\mathcal{O}(N^2_P \cdot N_A \cdot N_J)$	$\mathcal{O}(N^2_P \cdot N_A)$	$\mathcal{O}(N_X \cdot N_J)$
[proposed]	$\mathcal{O}(N^2_H \cdot N_A \cdot N_O)$	$\mathcal{O}(N^2_H \cdot N_A \cdot (N_{LS} + N_C))$	$(\sim 4Bytes/m^2)$

Finally, we provide a numerical result about the real delays obtained when computing the active strategy. A run of the active stategy, with $N_H = 19$ position hypotheses and $N_E = 40$ exploration particles, needs about 87 seconds in a standard 2GB RAM, 1.86-GHz Core 2 Duo PC, running Linux Ubuntu kernel 2.6.17. Only the 1.8% of this delay is for the generation of the E set, which involves multi-hypothesis path planning. The 97% of this execution time has been to compute the synthetic observations, $o^s_1(e^m_{kj})$ and $o^s_1(e^m_{ki})$ of the laser scanner. Authors are confident that the proposed implementation can be fairly optimized to reduce these delays.

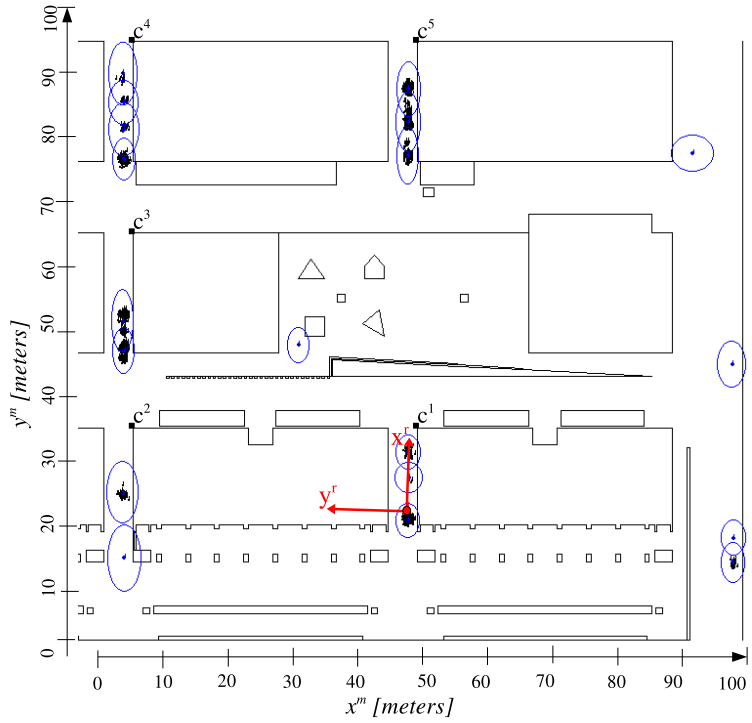

Fig. 8. The set H of $N_H = 19$ position hypotheses on the map. Each hypothesis is marked as a blue ellipse. Please note also the red dot marking the true robot position X_r^m, the robot frame (x^r, y^r) and the position of the $N_C = 5$ cameras.

8 SIMULATION RESULTS

8.1 Single Lost Robot. Non Cooperative vs Cooperative Environments

In this section we present the results of the active strategy for the single lost robot case, comparing the non cooperative environment of section 3 with the cooperative environment of subsection 4.1. The methodology has been as follows: the robot is placed in a given position and the particle filter is run using only the onboard robot sensors. For the presented run, this step has generated a set H of $N_H = 19$ position hypotheses showed in figure 8. Afterwards, an exploration particle set, E, is generated for both cases, obtaining a common set of $N_E = 40$ exploration particles, $\{e_1^r \dots e_{40}^r\}$. The evaluation of each exploration particle is performed separately, that is, in the non cooperative case, only the onboard sensors are considered in equation 9, while in the cooperative case also the camera network has been considered to evaluate the actions.

Figure 9 shows the E set of $N_E = 40$ exploration particles in the robot frame. We can see how the exploration set has been adapted to the multi-hypotheses constraints, thanks to the multi-hypotheses path planning step. The figure also shows, marked with a blue cross, the six best actions for the non cooperative

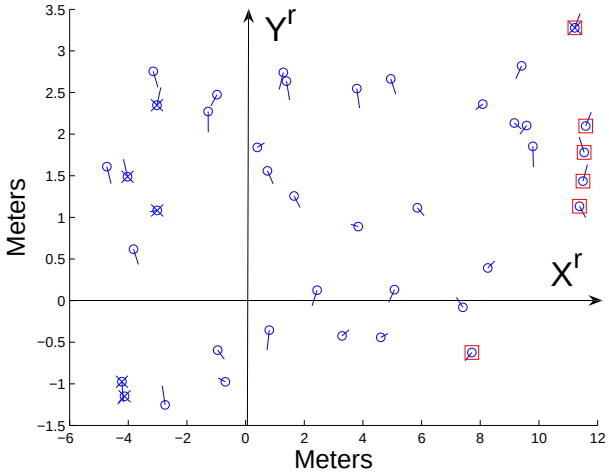

Fig. 9. The set E of $N_E = 40$ exploration particles that has been evaluated for both cases. The six best actions for the non cooperative case are marked with a blue cross and the six best ones for the cooperative case with a red square.

case, and, with a red square, the six best actions for the cooperative case. The reader can overlap the robot frame of this figure with the depicted robot frame on figure 8 in order to imagine where the robot will arrive if it goes to a given exploration particle. For the non cooperative case, the best actions are going to places which are distinctive from the laser scanner observation point of view. These six best actions are mainly related in going down of the corridor (see figure 8), since laser scanner is expected to take less ambiguous observations. On the other hand, for the cooperative case, the six best actions are going up of the corridor, since this is the area where more camera detections are expected taking into account all the hypotheses.

Figure 10 shows the value of each $\hat{N}(e_k^r),\ \forall k = 1 \dots 40$, for both, non cooperative and cooperative cases. This figure shows how the expected number of hypoteses is always bounded to $[1, N_H]$. We can observe that, for the cooperative case, the expected reduction of hypotheses is more significant that the obtained for the non cooperative environment. Therefore, some particles have the same $\hat{N}(e_k^r)$, indicating that no camera detection was expected even though considering all hypotheses, thus the onboard sensors remain the only means to disambiguate the situation. However, some other actions take benefit from the potential remote observations of the camera network and reduce clearly their evaluation index.

8.2 Cooperative Environment. Two Lost Robots

This section presents the results for the two lost robots case studied in subsection 4.2. In order to better evaluate this case, only the laser scanners of

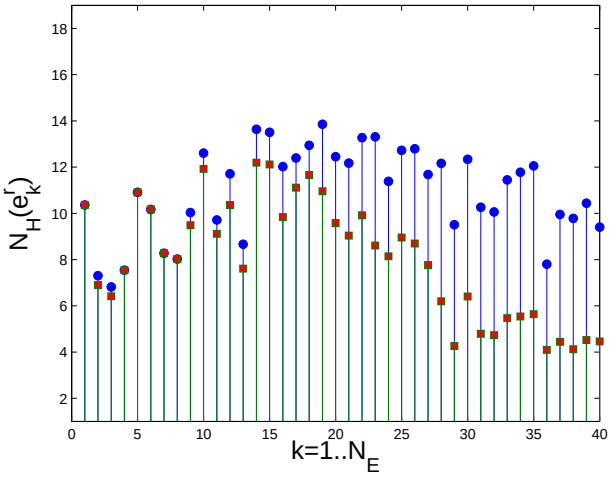

Fig. 10. The $\hat{N}(e_k^r)$ values of the exploration set E for both the non cooperative (blue dots) and the cooperative (red squares) cases. Particles are sort by its x^r coordinate in order to be related with figure 9.

the robots are considered in the active strategy, but there are no limitations on integrating remote observations of a camera network. We have proceeded placing two robots in the environment in positions within a line of sight. Robots first have performed a relative localization observation between them and then, have built the cooperative frame. Then the hypotheses generation step is performed with the particle filter, using the observations provided by the compasses and the laser scanners of each robot. For this case the particles s_i represent positions of the cooperative frame and this fact implies that each particle should satisfy a line of sight constraint imposed by the relative localization. Moreover, when position particles have to be corrected with real observations (see equation 22), they have to be translated to the relative positions of each observer given the observed line of sight. Thanks to this line of sight constraint and to the integration of more observations taken from two diferent points of view, the hypotheses generation step is improved and a few number of hypotheses generated. In the experiment presented in figure 11, where the two robots are placed at the same corridor as the one in figure 8, the number of generated hypotheses has been $N_H = 11$. The two lost robots case entails, at this initial step, an improvement in the hypotheses generation which implies less computational efforts to perform the active strategy, since time complexity depends on N_H^2.

Once the set H has been cooperatively generated, the active strategy is executed and outputs two joint actions, one to be performed by each robot. Figure 12 shows the E set generated in the presented execution and marks the selected primary action with a green square, the five best support actions considering the quantitative criterion with red diamonds and the five best support actions considering the qualitative criterion with green crosses. We

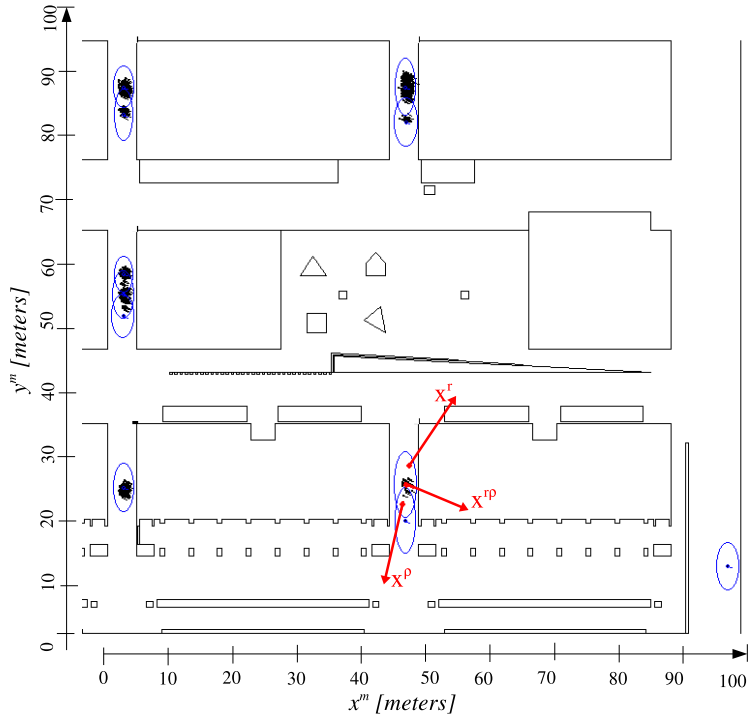

Fig. 11. The set H of $N_H = 11$ position hypotheses of the cooperative frame. Each hypothesis is marked as a blue ellipse. The x axis of each robot and that of the cooperative frame are also showed.

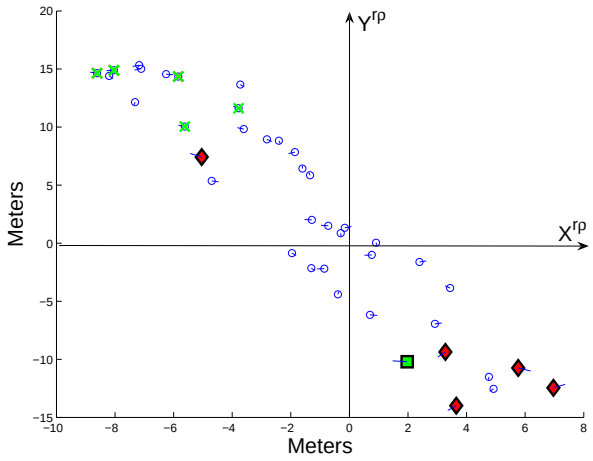

Fig. 12. The set E of $N_E = 40$ exploration particles that has been evaluated, in the cooperative frame. The green square marks the primary action, $e^{r\rho}_{s1}$. Five best secondary actions following the quantitative criteria (red diamonds). Five best secondary actions following the qualitative criteria (green crosses).

can see how the quantitative criterion would select redundant actions in most of the cases while the qualitative one clearly selects complementary actions.

9 CONCLUSIONS

Global localization is an important challenge for the mobile robot research community. We have presented in this paper a general theoretical probabilistic approach to solve the map-based global localization problem in large environments. This has been done by proposing an active strategy based on estimating the reduction factor of position hypotheses. The method is general since it is neither sensor dependent nor spatial representation dependent. It is based on some assumptions clearly exposed in section 2 that are, however, usually stated in map-based navigation approaches used in mobile robotics, such as the need of observation models. However, there are some contributions of this work we would like to pay attention to. The presented algorithm is computationally efficient, both in time and memory, when compared to other existing approaches, as it has been stated in section 7. Also note that, the presented active strategy selects the action to perform among a generated set of valid actions for the current situation of the robot, instead of selecting an action within a finite and fixed set of commands. However, the most important contribution of this work is the possibility to use the active strategy in cooperative environments, both for a single robot using a sensor network and for multiple robots within a sensor network context. This is an active and prominent research field within the robotics community.

An implementation of the proposed strategy has been proposed in order to present some simulated results in a large and real environment for the non cooperative and the cooperative cases. Single robot simulations show that the proposed method generates feasible commands to be executed by the robot in order to reduce the hypotheses set for the map-based global localization problem in large environments. The generated commands have the aim to drive the robot to the most distinctive areas of the environment, distinctive from the point of view of the onboard exteroceptive sensors. When a single lost robot is member of a robot network system, the results evidence how the behaviour of the robot changes taking benefits of the potential observations done by a camera set deployed on the environment.

For the two lost robot case, we establish a rational criteria to perform two joint actions when this robots are in a line of sight. These actions are choosen following a qualitative criteria that evaluates the level of redundancy of the actions. Results show how this criteria selects actions that are complementary form the point of view of the expected observations. Moreover, the two lost robot case benefits from a more accurate hypotheses generation step, since generated hypotheses to localize the cooperative frame integrates the observations of the two lost robots, and satisfies the line of sight constraint imposed by the relative localization observation.

References

[1] K.O. Arras, J.A. Castellanos, M. Schilt, and R. Siegwart. Feature-based multi-hypothesis localization and tracking using geometric constraints. *Journal of Robotics and Autonomous Systems*, 44:41–53, 2003.

[2] A. Corominas Murtra and J.M. Mirats Tur. Map Format for Mobile Robot Map-based Autonomous Navigation. Technical Report IRI-DT 2007/01, Institut de Robòtica i Informàtica Industrial, Barcelona, Spain, February 2007.

[3] Y. Dai and G. Yan. Collaborative Localization of Multi Micro-Robots Based on Markov Algorithm. In *Proceedings of the International Symposium on MicroMechatronics and Human Science*, 2003.

[4] D. Filliat and J-A. Meyer. Map-based Navigation in Mobile Robots. I. A review of localization strategies. *Cognitive Systems Research*, 4, December 2003.

[5] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. A Probabilistic Approach to Collaborative Multi-Robot Localization. *Autonomous Robots. Special Issue on Heterogeneous Multi-Robot Systems*, 8(3):325–344, June 2000.

[6] D. Fox, W. Burgard, and S. Thrun. Active Markov Localization for Mobile Robots. *Robotics and Automous Systems*, 25(3-4):195–207, 1998.

[7] D. Fox, W. Burgard, and S. Thrun. Markov Localization for Mobile Robots in Dynamic Environments. *Journal of Artificial Intelligence Research*, 11:391–427, 1999.

[8] B. Gärtner. Fast and robust smallest enclosing balls. In *Proceedings of the European Symposium on Algorithms (ESA)*, Prague, Czech Republic. July, 1999.

[9] A. Gasparri, S. Panzieri, 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] P. Jensfelt and S. Kristensen. Active Global Localisation for a Mobile Robot Using Multiple Hypothesis Tracking. *IEEE Transactions on Robotics and Automation*, 17(5), October 2001.

[11] S.M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[12] 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.

[13] J. Maantay and J. Ziegler. *GIS for the urban environment*. Environmental Systems Research Institute, 2005.

[14] R. Madhavan, K. Fregene, and L.E. Parker. Distributed Heterogeneous Outdoor Multi-robot Localization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC. May, 2002.

[15] E. Menegatti, M. Zoccarato, E. Pagello, and H. Ishiguro. Image-based Monte Carlo localisation with omnidirectional images. *Journal of Robotics and Autonomous Systems*, (48), 2004.

[16] G. Metta, P. Fitzpatrick, and L. Natale. YARP: Yet Another Robot Platform. *International Journal on Advanced Robotics Systems*, 1(3):43–48, 2006.

[17] J.M. O'Kane. Global localization using odometry. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, USA. May, 2006.

[18] J.M. Porta, J.J. Verbeek, and B.J.A. Kröse. Active Appearance-Based Robot Localization Using Stereo Vision. *Autonomous Robots*, 18(1):59–80, 2005.

[19] S. Premvuti and J. Wang. Relative position localizing system for multiple autonomous mobile robots in distributed robotic system : System design and simulation. *Journal of Robotics and Autonomous Systems*, 18(3), August 1996.

[20] 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.

[21] I.M. Rekleitis, G. Dudek, and E.E. Milios. Graph-Based Exploration using Multiple Robots. In *Proceedings of the International Symposium on Distributed Autonomous Robotic Systems (DARS)*, Knoxville, Tennessee, USA. October, 2000.

[22] I.M. Rekleitis, G. Dudek, and E.E. Milios. Multi-Robot Cooperative Localization: A Study of Trade-offs Between Efficiency and Accuracy. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Lausanne, Switzerland. September, 2002.

[23] S.I. Roumeliotis and G.A. Bekey. Distributed Multi-Robot Localization. *Transactions on Robotics and Automation*, 18(5):781–795, October 2002.

[24] 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.

[25] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust Monte Carlo localization for mobile robots. *Artificial Intelligence*, 128:99–141, 2001.

[26] 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.