Engineering Applications of Artificial Intelligence ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

A hybrid solution to the multi-robot integrated exploration problem

Miguel Juliá *, Óscar Reinoso, Arturo Gil, Mónica Ballesta, Luis Payá

Systems Engineering Department—Miguel Hernández University, Avda. Universidad s/n. Edif. Quorum V, 03202 Elche (Alicante), Spain

ARTICLE INFO

Article history:
Received 21 January 2009
Received in revised form
7 September 2009
Accepted 9 December 2009

Keywords:
Integrated exploration
Hybrid architecture
Cooperative robots
Expected safe zone
Gateway cell
Exploration tree

ABSTRACT

In this paper we present a hybrid reactive/deliberative approach to the multi-robot integrated exploration problem. In contrast to other works, the design of the reactive and deliberative processes is exclusively oriented to the exploration having both the same importance level. The approach is based on the concepts of *expected safe zone* and *gateway cell*. The reactive exploration of the *expected safe zone* of the robot by means of basic behaviours avoids the presence of local minima. Simultaneously, a planner builds up a decision tree in order to decide between exploring the current *expected safe zone* or changing to other zone by means of travelling to a *gateway cell*. Furthermore, the model takes into account the degree of localization of the robots to return to previously explored areas when it is necessary to recover the certainty in the position of the robots. Several simulations demonstrate the validity of the approach.

1. Introduction

Exploration is the task of covering an unknown area by a mobile robot or a group of robots. Usually, they build a model of the environment at the same time. Some applications of exploration are automated surveillance, search and rescue services or map building of unknown environments as, for example, in planetary missions. Compared to the case of a single robot, the utilization of a team of cooperative mobile robots is an advantage (Cao et al., 1997; Farinelli et al., 2004): the exploration time is reduced and the precision of the maps is improved because of the redundancy of measurements (Rekleitis et al., 1997, 2001).

As stated by other authors (Stachniss et al., 2005b; Makarenko et al., 2002), the exploration problem is related to the mapping and localization tasks. Fig. 1 shows this relation and the algorithms that resolve these different problems:

- *Simultaneous localization and mapping* (SLAM) algorithms are used to create a map of the environment and to simultaneously localize the robots in it.
- *Classic exploration* algorithms decide the best movements to guide the robot to quickly create a map of the environment.
- *Active localization* algorithms guide the robots to the best positions to achieve a good localization.
- *Integrated exploration* algorithms decide the movements of the robots in order to create a map while minimizing the error in the trajectories and the obtained map.

Generally, SLAM techniques are employed simultaneously with classic exploration algorithms (Simmons et al., 2000). However, the result obtained by the SLAM algorithm strongly depends on the trajectories performed by the robots (Stachniss et al., 2005b; Makarenko et al., 2002). Classic exploration algorithms do not take localization uncertainty into account and direct the exploration in order to minimize the distance travelled while maximizing the information gained. When the robots travel through unknown environments, the uncertainty over their position increases and the construction of the map becomes difficult. Consequently, the result could be a useless and inaccurate map. Returning to previously explored areas or closing loops reduces the uncertainty over the pose of the robots and improves the SLAM process. This idea is commonly denoted as *integrated exploration* or SPLAM (simultaneous planning localization and mapping). With this technique the robots explore the environment efficiently and also consider the requisites of the SLAM algorithm.

The goal of this paper is to develop an integrated exploration algorithm. We will have to come to an agreement between the speed of exploration and the quality of the generated maps. At the same time, the algorithm must work in real time and it must be robust, thus we need a decentralized approach. One of the problems in exploration and map building is the dependence of the computational time of the exploration algorithm on the dimension of the map. For this cause, the objective of real-time processing can be difficult to achieve if we are working with large

* Corresponding author. Tel.: +34 965222435; fax: +34 966658979.
E-mail addresses: mjulia@umh.es (M. Juliá), o.reinoso@umh.es (Ó. Reinoso), arturo.gil@umh.es (A. Gil), m.ballesta@umh.es (M. Ballesta), lpaya@umh.es (L. Payá).

0952-1976/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.engappai.2009.12.005

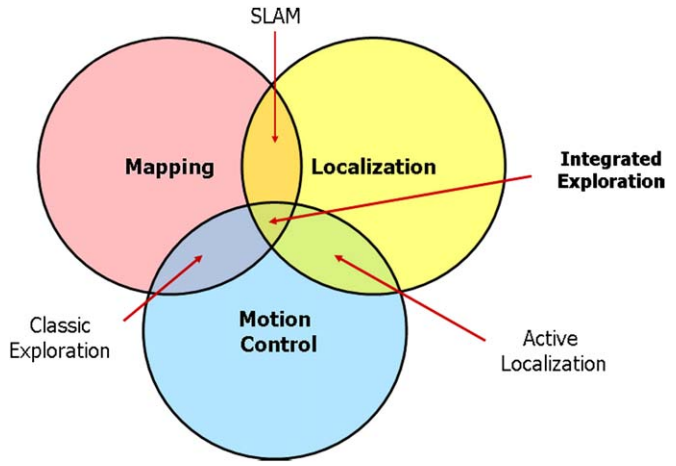

Fig. 1. The figure shows the algorithms that implement the mapping, localization and motion control task in the exploration problem. Integrated exploration algorithms decide the movements that quickly create a map while minimizing the error in the trajectories and the obtained map.

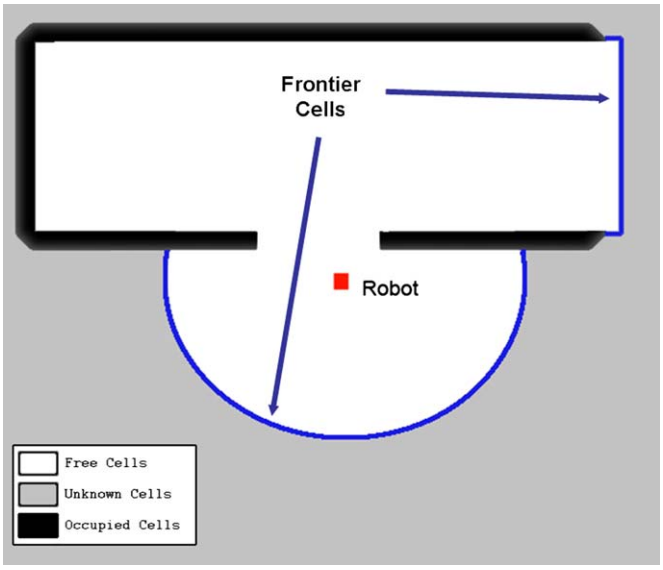

Fig. 2. The figure shows an occupancy grid map. The grey level of each cell indicates the occupation probability. The frontier cells, defined as the free cells next to an unknown cell, are emphasized in the graphic.

maps. In this sense, the algorithms should be independent of the dimensions of the map. For this cause, the algorithm we propose in this paper allows a robust integrated exploration because of the decentralization and the use of local maps that reduces the processing time.

Therefore, in this paper we present a hybrid solution to the multi-robot integrated exploration problem. Section 2 presents the state of the art in the field of exploration. Section 3 defines the main ideas of the approach and explains the advantages of the developed model. In Section 4 the proposed approach is explained in detail. Section 5 presents the experiments that were carried out to test the method and their results. In Section 6 our technique is compared with other integrated exploration techniques. Finally, the main conclusions are exposed in Section 7.

2. Related work

Typically, exploration techniques work basically using the frontier concept introduced by Yamauchi (1997). In a regular grid map that represents the occupation probability, as introduced by Moravec and Elfes (1985), cells can be classified as free, occupied or unknown. This information can be obtained by any kind of range sensor. Using this sort of map, Yamauchi defined the frontier cells as free cells that lie next to an unknown cell. We can see an example of occupancy grid map in Fig. 2, where the frontier cells are emphasized. Most of the exploration techniques use an occupation probability map and the frontier concept. However, there are other approaches that use other forms of identifying the regions of interest for the exploration. For instance, Wullschleger et al. (1999) and Newman et al. (2003) perform the exploration by means of directing the robots to open segments or features of the map, Murphy and Newman (2008) use a gap navigation directing the robots to the occluded zones of the sensor, and Santosh et al. (2008) lead the robots to the limits of the floor detected in images using only visual information.

Focusing on the exploration planning, we can distinguish two types of approaches to the exploration problem: deliberative and reactive.

The group of deliberative exploration methods usually employs path planning techniques (Fernandez et al., 1999) in order to direct the robots to the frontier cells. They differ in the coordination strategies used to assign a destination to each robot. A basic strategy is that the robots go to the nearest frontier as in the work of Yamauchi (1998). A cost-utility model has been also used to decide good destinations in single-robot exploration (Gonzalez-Baños and Latombe, 2002; Amigoni, 2008). In this sense, some authors have extended this kind of model to coordinate the robots (Simmons et al., 2000; Stachniss et al., 2006; Burgard et al., 2005). Normally, the cost is the length of the path to a frontier cell, whereas utility can be understood in different ways: Simmons et al. (2000) consider the utility as the expected visible area behind the frontier, Stachniss et al. (2006) use semantic information to increase the utility of the candidate destinations situated in corridors. With a higher level of coordination, Burgard et al. (2005) consider in the utility function the proximity to frontiers that were previously assigned to other robots. This way, the exploration speeds up since the robots choose different frontiers that are far from each other. Some authors include other types of representations of the environment in their approaches. For instance, Franchi et al. (2007) make the planning over a sensor-based random tree (SRT). The tree is expanded as new candidate destinations near the frontiers of the sensor coverage are selected, and it is used to navigate back to past nodes with frontiers when no frontiers are present in the current sensor coverage. In a similar way, Rocha et al. (2008) selects the best frontier from the current sensor coverage and uses also a topological map when there are no visible frontiers. Other approaches focus on the structure of the environment. The doors that divide the environment in corridors and rooms can be identified and represented in a topological map. Wurm et al. (2008) take advantage of this information for assigning optimally a different unexplored room to each robot using the *Hungarian method*. Other authors approach this issue as the *travelling salesman problem* by means of optimizing a complete route for the robots having each robot an ordered sequence of frontiers to visit. In this sense, Zlot et al. (2002) suggest using a market economy where the robots optimize their routes by means of negotiating their destinations.

The other group of exploration techniques is reactive and commonly they are behavioural approaches (Arkin and Diaz, 2002; Lau, 2003; Juliá et al., 2008; Schmidt et al., 2006). The combination of a set of behavioural forces points out the advance direction. Arkin and Diaz (2002) combine an *avoid obstacles*

Please cite this article as: Juliá, M., et al., A hybrid solution to the multi-robot integrated exploration problem. Engineering Applications of Artificial Intelligence (2010), doi:10.1016/j.engappai.2009.12.005

behaviour with an *avoid past* behaviour, that generates a force directed away from previously visited areas, and a *Probe* behaviour, that directs the robots to free space. At the same time, to preserve communication, they include behaviours to maintain a free line of sight between the robots. Lau (2003) employs a *move to frontier*, an *avoid obstacles*, and an *avoid robots* behaviours. This leads to the avoidance of collisions and also improves the exploration by dispersing the robots. As stated by many authors, the main drawback of this technique is the occurrence of local minima in the potential field, which may trap the robot and block the exploration process. In this sense, a technique to detect these situations, by means of analysing the potential field generated by the combination of behavioural forces in the proximities of the robot, is shown in Juliá et al. (2008). Once the local minimum is detected, a technique to force the robot to escape from this point is necessary. A solution can be to plan a path to a frontier cell (Lau, 2003; Juliá et al., 2008). A more reactive but less efficient solution is using a wall-following strategy (Xiaoping and Ko-Cheng, 1997). Harmonic functions are also used to preform a control free of local minima (Prestes et al., 2002), however, this technique is computationally expensive as it needs to evaluate a global potential field. Garrido et al. (2008) use a similar technique based on the Voronoi Fast Marching method. Schmidt et al. (2006) use a different set of behaviours: *hold distance to obstacle*, *local door driving*, *narrow driving*, *random cruise* and *pre evasion*. This behavioural system is combined with the information of a topological map to decide the current behaviour.

These previously cited classic exploration algorithms do not take localization uncertainty into account and direct the exploration in order to minimize the distance travelled while maximizing the information gained. If we do not consider the uncertainty in the position of the robot, the construction of the map could be difficult and the result could be an useless and inaccurate map. When we take the localization uncertainty into account in the exploration algorithm we talk about *integrated exploration algorithms*. The main idea is that we have to consider other possible destinations besides frontiers. Going to previously explored zones or closing loops in the environment are useful actions to reduce the uncertainty. At the same time, not all the frontiers will be reachable with the same uncertainty in the position of the robots. From the point of view of localization, it is desirable to travel to frontiers that are situated close to precise landmarks of the environment. Some algorithms of this type has been previously developed by other authors (Makarenko et al., 2002; Stachniss et al., 2005a; Freda et al., 2006; Juliá et al., 2008; Tovar et al., 2006). Makarenko et al. (2002) include the uncertainty in the localization as part of the utility function in the assignment of destinations to robots in a cost-utility path planning approach. In this way, the frontiers that are near to landmarks has a higher utility. Stachniss et al. (2005a) consider three types of possible destinations: frontiers, past poses, and points that close a loop (Stachniss et al., 2005b). These destinations are evaluated considering the expected information gain when travelling to that point. This information gain may come from the incorporation of new zones to the map or from reducing the uncertainty of the map. Freda et al. (2006) use a sensor-based random tree (SRT) in which the candidate destinations to expand the tree are analysed considering the reliability of the expected observable features from that points. In Juliá et al. (2008), using a behavioural approach, a *Go to precise landmarks* behaviour is used to send the robots back to previously explored areas. Tovar et al. (2006) considers in a utility function the number of landmarks that are observable in a path to potential targets near the frontiers. These potential targets are evaluated in a decision tree considering the utility of being reached from the different robots of the team in first term or after visiting other destinations.

Despite the fact that the typical architectures used are commonly hybrid deliberative/reactive approaches (Posadas et al., 2008), the exploration is mainly carried out by one of these processes using one of the approaches previously cited. In this sense, we talk about a deliberative exploration technique when the exploration is planned by the deliberative process while a low level reactive process commands the robot safely. And we talk about a reactive exploration technique when the exploration is carried out by means of including simple reactive exploration behaviours as, for instance, *go to frontiers*. However, reactive exploration architectures have usually a very simple high level process that triggers some configurations for the behaviours, sometimes combining the exploration with other tasks.

In contrast to other related works, in our hybrid approach the deliberative and the reactive processes are exclusively designed to explore, having both the same importance level. In our architecture, the reactive process is able to carry out a short term exploration while the deliberative process plans a long term exploration. Moreover, both processes include multi-robot coordination mechanisms. This way, our algorithm has the main advantages of both techniques. Furthermore, our technique includes mechanisms for returning to previously explored zones performing an *integrated exploration*. In this pure hybrid sense only the exploration approach exposed in Schmidt et al. (2006) has been developed, however, it does not perform an *integrated exploration* and it is a mono-robot approach that does not provide coordination for several robots. In Lau (2003) and Juliá et al. (2008) we can also see a reactive exploration technique that uses some deliberation in the auxiliary path planner embedded to scape from the local minima, but this deliberation is used as an auxiliary state to solve the local minima problem.

3. Reactivity and deliberation: the hybrid approach

As we said, the main drawback of the reactive methods is the appearance of local minima. The origin of these points, where the robots get blocked, is mainly the presence of points of attraction behind the obstacles. Fig. 3 shows three common situations that present a local minimum when using an *avoid obstacles* repulsive behaviour. Usually, the solution to this problem consists in planning a path to a frontier cell when a local minimum is detected near the robot. Being able to detect and escape from these situations avoids that the robots get blocked by the local minima and thus the exploration process can be finished. This type of reactive system has been proved as a valid approach to multi-robot exploration (Lau, 2003; Juliá et al., 2008). However, local minima have a negative effect in the performance of the exploration algorithm because of the waste of time in the process

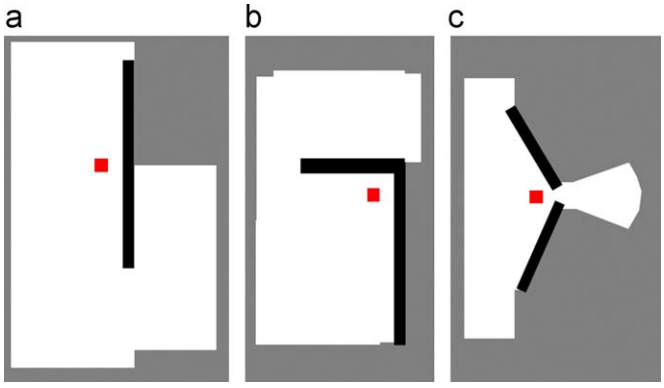

Fig. 3. The figure shows three different local minimum situations: (a) long planar obstacle with a frontier behind, (b) concave obstacle and (c) narrow way.

Please cite this article as: Juliá, M., et al., A hybrid solution to the multi-robot integrated exploration problem. Engineering Applications of Artificial Intelligence (2010), doi:10.1016/j.engappai.2009.12.005

of travelling to the local minimum, detecting it and planning a new route to escape from that point.

We can see that this sort of reactive approach performs well only in the proximities of the robots when no local minima are present. However, in order to travel a long way to a far frontier when local minima are likely to appear and having some information of that zone in the map, a deliberative method is better. These facts lead us to a hybrid reactive/deliberative architecture. In contrast to other related works, in our approach the reactive and deliberative processes are exclusively designed to explore and they perform a local minima free exploration.

The first step to consider is how to avoid local minima in the reactive process. An option is using a tangential force in the *avoid obstacles* behaviour that makes the robot follow the walls (Xiaoping and Ko-Cheng, 1997). But, without analysing deliberatively which is the best direction to continue with the exploration, we cannot decide properly in which direction the robot must go round the obstacle. If we select a random direction, we would avoid minima but at the cost of loosing efficiency.

For this cause, we have designed a better solution that consists in not considering the cells behind the obstacles. In this sense, only the cells in a free of obstacles view from the robot affect the reactive process. Thus, the reactive process is blind to the cells behind the obstacles, letting the deliberative process to consider them. This way, we introduce a sense of spatial connectivity in the model. In order to understand how the whole hybrid model works we have to make some definitions:

- *Expected safe zone* of a robot is the set of free or unknown cells that can be joined with the position of the robot with a straight line without intersecting any detected obstacle until a maximum distance (d_{esz}).
- *Gateway cell* is any free cell within the expected safe zone of a robot next to a free cell not belonging to this zone.

These two concepts are shown graphically in Fig. 4. The *expected safe zone* covers, like a virtual sensor over the map, the surroundings of the robot until the distance d_{esz} or founding an obstacle. As we can see, the distance d_{esz} is greater than the sensor range d_s and has no relation with the real sensor. The real sensor could be directional and cover only a limited angular arc but the *expected safe zone* covers the 360°.

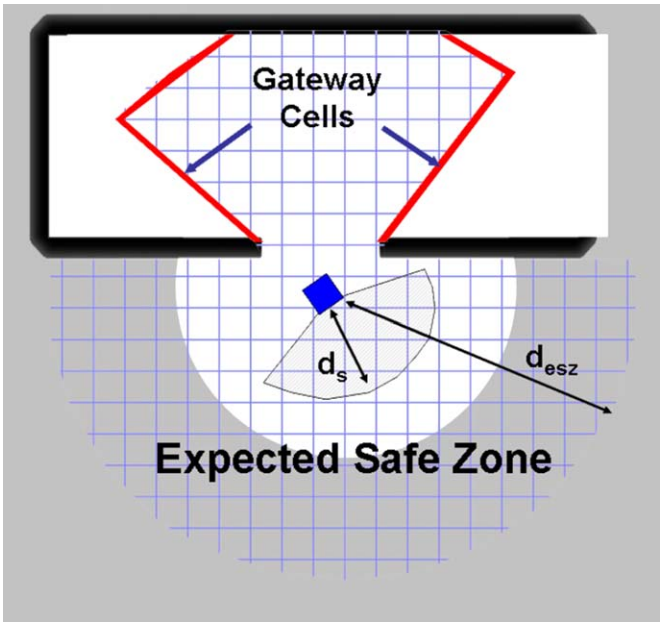

Fig. 4. Expected safe zone and gateway concepts. The expected safe zone covers the surroundings of the robot in 360° until the distance d_{esz} or finding an obstacle. The gateways cells within the expected safe zone limit with free cells not belonging to the expected safe zone.

Basically, our new approach consists in the reactive exploration of the *expected safe zone* in the proximities of the robots. In this way, we avoid the appearance of local minima as the cells behind the obstacles are not considered. Simultaneously, a planner evaluates when to travel to other zones navigating to a *gateway cell*.

The concept of *safe zone* has been previously used in exploration algorithms (Gonzalez-Baños and Latombe, 2002; Franchi et al., 2007), but it is used in connection with the range of the sensor. We have use the expression *expected safe zone* because the zone that we have defined covers a larger area than the sensor, therefore it includes unexplored areas.

The movements of the robots will be evaluated using a two layers system. The reactive layer is the combination of several basic behaviours that include common behaviours as *go to frontier*, *avoid obstacles* or *go to gateway*. This layer operates only with cells within the *expected safe zone*. The deliberative layer controls the reactive layer enabling and disabling behaviours and setting the gateways. Thus, the deliberative layer is able to switch between several states or combination of behaviours. We work with these three states that are explained in the next section:

1. explore current expected safe zone;
2. change zone;
3. active localization.

The main task of the deliberative layer is to decide the current state. These three states are designed for integrated exploration, taking into account the uncertainty over the position of the robots. In this sense, the *active localization* state leads the robot to past positions to recover the localization. The deliberative layer makes the decision between *exploring the current expected safe zone*, travelling to past poses using the *active localization* state or travelling to a gateway (*change zone*). This decision is made by considering the uncertainty in the pose of the robots and the analysis of an exploration decision tree that will be explained in the next section. In this tree, the root node represents the current *expected safe zone* and the branches represent the *gateways* to other zones. The tree generated is similar to the SRT method (Franchi et al., 2007), but they differ in the use of the *expected safe zone* instead of the sensor *safe zone* and in the fact that our exploration tree is built exclusively to plan instead of being a register of past states. Furthermore, the leaves of the tree are situated over the frontiers incorporating the expected information gain from each frontier (as in Simmons et al., 2000).

The approach explained in this paper presents some advantages. First, the avoidance of the local minima reduces the exploration time. Second, the reactive process runs in a delimited period of time because of the delimitation of the map considered to the *expected safe zone*. In this way, the process time of the exploration algorithm (at least the reactive critic task) does not depend strongly on the size of the map and thus we are satisfying the objective of real-time processing. Finally, this delimitation of the map considered to the local surroundings of the robot presents an advantage in the robustness of the algorithm. This is because it is relatively easy to work with this approach in a distributed manner. Avoiding a centralized approach the algorithm becomes more robust, and this is possible if the reactive system is able to run with a local map only.

We are now using a centralized SLAM technique, but it can be separated using each robot his own map. The reactive layer only

Please cite this article as: Juliá, M., et al., A hybrid solution to the multi-robot integrated exploration problem. Engineering Applications of Artificial Intelligence (2010), doi:10.1016/j.engappai.2009.12.005

uses the local *expected safe zone* map, and there is no need for each robot to know and manage the global map. However, the deliberative layer needs information of the global map. It would be necessary to know the alignment between the maps (Ballesta et al., 2008b) and to provide certain mechanisms of coordination between the robots using a communication channel. The use of this sort of totally distributed system is desirable because there is no need for knowing the initial positions (Fox et al., 2006; Chao-xia et al., 2008) and thus the robots can begin the exploration in a dispersed way that is more efficient in relation to the exploration time. In this paper, we focus on the hybrid architecture of the system and the distribution of the SLAM algorithm is postponed to future works.

4. Architecture

We can distinguish three parts in the designed architecture: a centralized SLAM, which builds the maps and obtains the localization, a deliberative layer, and a reactive layer. Fig. 5 shows this model. It consists of the centralized SLAM process and two processes per each robot running concurrently: one deliberative and one reactive. All these processes run concurrently as independent threads, that is, a thread for the common SLAM and two threads per each robot. Reactive and deliberative processes have access to the maps created by the SLAM process. The deliberative layer in each robot controls the reactive layer, enabling and disabling states or combinations of behaviours and setting the gateways. Next we explain these three parts.

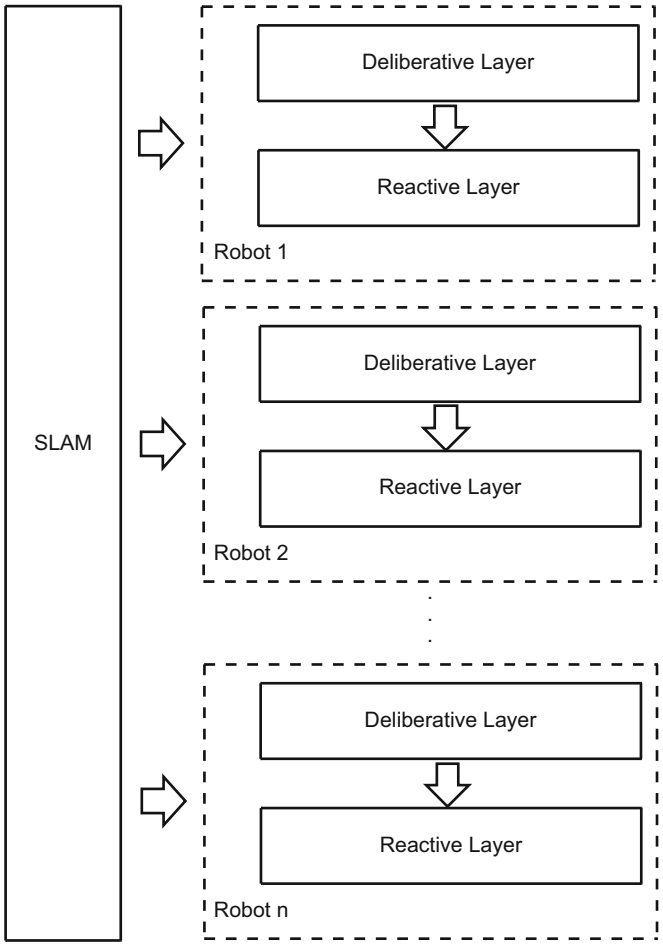

Fig. 5. The figure shows the architecture of the hybrid multi-robot exploration system. It consists of a centralized SLAM process and two processes per each robot running concurrently: a deliberative process and a reactive process. Reactive and deliberative processes have access to the maps generated by the SLAM process, besides, the deliberative process is able to configure the reactive process.

4.1. SLAM

In typical environments we can find a set of highly distinctive elements that can be easily extracted with the sensors of a robot. These elements are typically called landmarks. In our application, we assume that the robots are able to detect a set of distinctive 3D visual landmarks and they are able to obtain relative measurements to them using stereo cameras. These landmarks can be extracted as interest points found in the images of the environment (Ballesta et al., 2008a). The robot team is able to build a map with a vision-based technique (Gil et al., 2010) consisting in a particle filter approach to the SLAM problem which is known as FastSLAM (Montemerlo and Thrun, 2003).

However, landmark based maps do not represent the free or occupied areas in the environment. For this reason, we use an auxiliary low resolution grid map to represent the free, occupied or unknown state of the space using the information of a sonar (Moravec and Elfes, 1985). In that map we can identify frontier cells as free cells that lie next to an unexplored cell. It is also possible to run the SLAM directly over the occupancy grid map using a more precise range sensor as a laser, but this seems a more expensive solution. Anyway, both implementations of SLAM can be used with our integrated exploration algorithm.

Thus, we have a set of M pairs of maps (visual landmarks and occupancy grid map), one pair per each particle in the filter. These maps are associated to a determined path performed by the robots. One of the particles will be the most probable, and thus, its associated couple of maps will be the most probable.

An example of occupancy grid map was shown in Fig. 2. Fig. 6 shows an example of visual landmarks map. Each map is composed by the set of detected landmarks until that moment. For each landmark, it is saved its 3D position, its variance and a visual descriptor for correspondence.

The uncertainty in the localization of the robots can be estimated using the dispersion in the positions of the robots between the different particles of the filter. This dispersion is compared with two thresholds in a hysteresis loop in order to consider the robots as well or poorly localized. When the dispersion is higher than the high threshold, we consider the robot as bad localized. Although the dispersion was reduced,

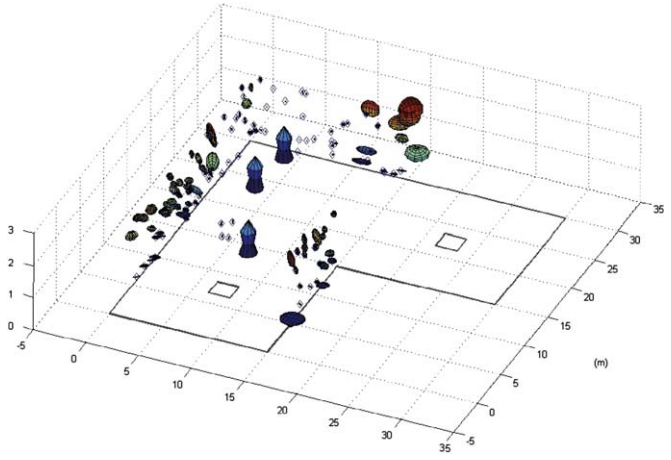

Fig. 6. The figure shows an example of visual landmarks map and the positions of three robots. Each landmark is marked in its 3D position with an ellipsoid with proportional dimensions to its variance.

Please cite this article as: Juliá, M., et al., A hybrid solution to the multi-robot integrated exploration problem. Engineering Applications of Artificial Intelligence (2010), doi:10.1016/j.engappai.2009.12.005

we will not consider the robot as well localized until the dispersion is below the low threshold. This hysteresis avoids a constant change between going to unexplored zones and returning to past poses.

In order to reduce uncertainty, we need to save the past poses of the robots when they are well localized. To register these poses we use a binary grid map to mark all positions where the dispersion had a value below the given low threshold for good localization. In this sense, when the dispersion is below that low threshold, we mark the cell in which the robot is situated for the most probable particle. This is a simple way to save good places where the robots may recover the localization certainty.

4.2. Reactive layer

The reactive layer is controlled by the deliberative layer. The deliberative layer sets the active combination of behaviours that runs in the reactive layer. The combination of the active behaviours determines the direction of the robot movement. The behaviours work only with the cells in the *expected safe zone*. Next, we explain the behavioural model used in the reactive layer and how to obtain the *expected safe zone*.

4.2.1. Behavioural model

Our approach to the problem of multi-robot exploration consists of six basic behaviours whose composition results in the trajectory of each robot in the environment:

Go to unexplored areas: Each unexplored cell attracts the robot.

Go to frontier: This behaviour attracts the robots to frontier cells since these are the cells that give way to areas of interest.

Avoid other robots: This behaviour results in a repulsive force between robots that normally allows to spread the robots around the environment.

Avoid obstacle: Each cell within a specific range that is identified as belonging to an obstacle applies a repulsive effort over the robot. This range allows to easily adjust the system.

Go to gateway: This behaviour attracts the robot to a *gateway cell* in order to access to other zones.

Table 1
Forces defined for each behaviour.

Go to unexplored areas:	$\vec{F}_k^1 = \frac{1}{M_U} \sum_{i=1}^{M_U} \frac{\vec{s}_i - \vec{p}_k}{r_{i,k}^3}$
Go to frontier:	$\vec{F}_k^2 = \frac{1}{M_F} \sum_{i=1}^{M_F} \frac{\vec{s}_i - \vec{p}_k}{r_{i,k}^3}$
Avoid other robots:	$\vec{F}_k^3 = \frac{1}{X} \sum_{j=1}^{X} -\frac{\vec{p}_j - \vec{p}_k}{r_{j,k}^3}$
Avoid obstacle:	$\vec{F}_k^4 = \frac{1}{M_O} \sum_{i=1}^{M_O} -\frac{\vec{s}_i - \vec{p}_k}{r_{i,k}^3}$
Go to gateway:	$\vec{F}_k^5 = \frac{\vec{q}_g - \vec{p}_k}{r_{g,k}^3}$
Go to precise poses:	$\vec{F}_k^6 = \frac{1}{M_P} \sum_{i=1}^{M_P} \frac{\vec{s}_i - \vec{p}_k}{r_{i,k}^3}$
M_U:	Number of unexplored cells
M_F:	Number of frontier cells
M_O:	Number of obstacle cells in the range
M_P:	Number of precise pose cells
X:	Number of robots
$\vec{s}_i$:	Position vector of the i-th cell
$\vec{q}_g$:	Position vector of the gateway g
$\vec{p}_j$:	Position vector of the j-th robot
$\vec{p}_k$:	Position vector of the k-th robot
$r_{i,k}$:	Distance from i-th cell to robot k
$r_{j,k}$:	Distance from robot j-th to robot k
$r_{g,k}$:	Distance from gateway g to robot k

Go to precise poses: This behaviour attracts the robot to cells marked as having low dispersion in the map.

Table 1 shows how the forces are calculated for each behaviour. Only the cells within the *expected safe zone* of the robots are used in the determination of each force. Fig. 7 shows the whole composition of the reactive model. As we can see, the composition of these basic behaviours is carried out taking into account a set of weights k_i whose value is experimentally deduced. Table 2 shows the weights selected for each behaviour. Besides, every behaviour can be enabled or disabled by the planner. The resulting force of the combination of these basic behaviours on each robot constitutes a vector that indicates the trajectory of the robot. In order to filter the movements over the discontinuities generated by the active range of the *avoid obstacles* behaviour we use the technique explained in Juliá et al. (2008). The final control command given to the motors in terms of linear and angular speed is calculated with a set of regulators that limit the linear and angular speed to suitable values for the SLAM algorithm.

4.2.2. Obtaining the expected safe zone

Obtaining the *expected safe zone* is essential in this approach. As we defined, only those cells inside the d_{esz} ratio that can be joined with a straight line with the robot position without intersecting any obstacle belong to the *expected safe zone*. That avoids all the local minima caused because of points of attraction behind the obstacles as in Fig. 3(a) or (b). Another sort of local minimum can appear when the point of attraction is visible but the robot must go across a too narrow way as in Fig. 3(c). To avoid this case, we have to process the *expected safe zone* to remove these zones behind a narrow way.

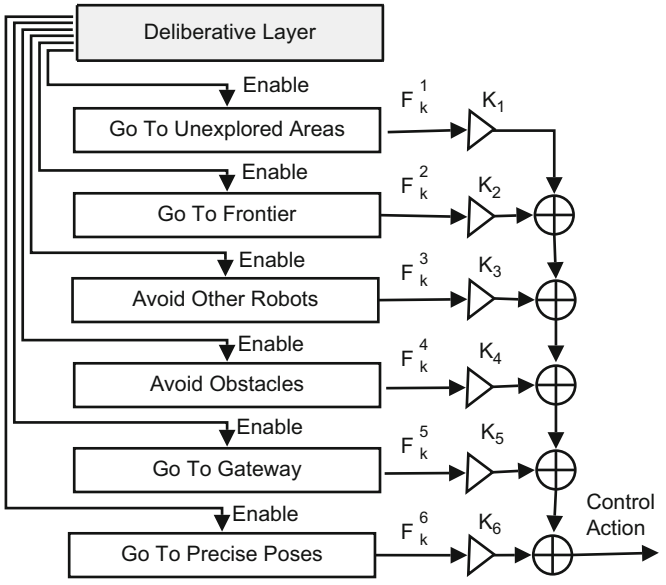

Fig. 7. The figure shows the reactive layer structure. It is composed by several basic behaviours that are enabled by the planner. The forces defined for each behaviour are linearly combined with a set of weights. The resultant force indicates the direction for the robot movement.

Table 2
Weights assigned to each behaviour.

k_1	k_2	k_3	k_4	k_5	k_6
5	15	0.25	100	1	0.5

Please cite this article as: Juliá, M., et al., A hybrid solution to the multi-robot integrated exploration problem. Engineering Applications of Artificial Intelligence (2010), doi:10.1016/j.engappai.2009.12.005

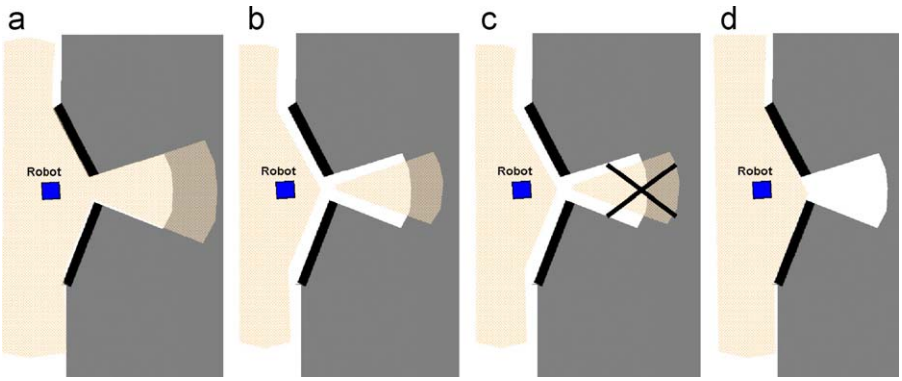

Fig. 8. The figure shows how to process the expected safe zone to avoid local minima in narrow ways: (a) incorporate the cells in the surroundings of the robot until a distance d_{esz} or finding an obstacle. (b) Erosion with depth relative to the avoid obstacle behaviour active range. (c) The separated regions are removed. (d) Dilatation of the remaining zone.

Table 3
Possible states for the reactive layer.

	State	Behaviours
1.	Explore current Expected safe zone	Go to unexplored areas Go to frontier Avoid other robots Avoid obstacle
2.	Change zone	Go to gateway Avoid obstacle
3.	Active localization	Avoid obstacle Go to precise poses

The process is shown in Fig. 8. These are the steps necessary to obtain the *expected safe zone*:

1. Select the cells up to a d_{esz} distance in a free of obstacles view from the robot.
2. Make an erosion over the obtained zone. The depth of the erosion must be related to the obstacle avoidance behaviour active range, which establishes the minimum distance to an obstacle.
3. If some region is separated with the erosion we take it off from the *expected safe zone*.
4. *Expected safe zone* is restored by means of dilating the remaining zone.

Only those cells in the resultant *expected safe zone* are considered by the reactive process. Obviously, in the *avoid obstacles* behaviour we consider only those obstacle cells next to the *expected safe zone*. In consequence, our reactive model is free from local minima.

4.3. Deliberative layer

The planner in the deliberative layer decides the state in which the reactive layer has to operate. This is made by means of enabling or disabling behaviours. The planner can also configure the gateway position for the *go to gateway* behaviour. We work with a very simple set of general states so that the solution is general enough. This way, we have a great independence of the considered environment.

Table 3 shows the three states and the behaviours that are enabled for each one. The first state consists in the reactive exploration of the current *expected safe zone*. The second state leads the robot to other zones, directing the robot to a gateway cell. The last state, *active localization*, attracts the robots to past poses recorded as having a good localization. The evaluation of an *exploration tree* decides the transition between these states.

The deliberative layer runs concurrently on each robot with its reactive layer in independent threads but it has less loop frequency than the reactive layer. In each iteration the planner generates a new exploration tree from the new position of the robot and decides the next state.

4.3.1. Creating the exploration tree

The first step in order to decide the next state is the creation of the *exploration tree*. The tree begins in the position of the robot. Each node has a spatial position with an associated zone and a cost. We can add two types of nodes to the tree: branches and leaves. We add a branch for each gateway we found. The leaves represent the objective of the exploration. In this sense, they are related with frontiers and precise cells. We would add a leaf for each frontier found or, if we have high position uncertainty, we would add a leaf for each group of precise cells. Algorithm 1 details the process of creation of the exploration tree.

Algorithm 1. Exploration tree creation algorithm

```
1:  Add root node to the tree in the position of the robot
2:  Associate the expected safe zone to the root node
3:  Look for gateways in safe zone→add branches
4:  if robot is well localized then
5:    Look for frontiers in safe zone→add leaves
6:  else
7:    Look for precise cells in safe zone→add leaves
8:  end if
9:  Add expected safe zone to processed zone
10: repeat
11:   Select next low cost branch
12:   New zone = expected safe zone from branch
13:   Take the processed zone away from new zone
14:   Associate new zone to branch
15:   Look for gateways in new zone→branches
16:   if robot is well localized then
17:     Look for frontiers in new zone→leaves
18:   else
19:     Look for precise cells in new zone→leaves
20:   end if
21:   Count robots in the new zone
22:   Add new zone to processed zone
23: until no remaining branches
24: for all leaf in tree do
25:   new zone = expected safe zone from the leaf
```

Please cite this article as: Juliá, M., et al., A hybrid solution to the multi-robot integrated exploration problem. Engineering Applications of Artificial Intelligence (2010), doi:10.1016/j.engappai.2009.12.005

```
26:      if robot is well localized then
27:        Count unexplored cells in new zone
28:      else
29:        Count precise cells in new zone
30:      end if
31:    end for
```

Fig. 9 shows an example of how to create the *exploration tree* during an exploration with four robots. Next, we explain the algorithm with that example.

First we add the root node to the tree with the current position of the robot. We have to determine the *expected safe zone* in this position. Next, we look for the gateway cells present in the *expected safe zone*. These gateway cells are grouped by proximity and a branch is added to the tree for each clustered gateway found. Then we proceed with the frontiers (or precise cells if bad localization) in the same way, but adding leaves. We can see in Fig. 9(a) how the tree begins in the position of the robot. The *expected safe zone* has been obtained and the doors and the frontiers have been localized in it. A branch, represented by a circle, has been added for each door and a leaf, represented by a square, has been added for each frontier.

Next, the branch with low cost in terms of distance travelled is selected and the operation explained above is repeated. It is important that the previously processed zone is subtracted from the new *expected safe zone*, which is evaluated from the position of the considered node, in order to expand the tree. Fig. 9(b) shows how the tree has been expanded, choosing the closer branch. The zone associated to this branch is the obtained as the *expected safe zone* from this point taking the previously processed zone away from it. A new frontier is found in this zone and a new leaf is added consequently.

This process continues with the next low cost remaining branch until no new gateways are found. The costs are accumulated from a branch to the derived nodes. Every time we get a new zone, we have to check if there is any robot in it and save the number of robots for the evaluation. In Fig. 9(c) we can see the full developed tree. As shown, there are robots in some zones.

Finally, for each leaf node, we have to count the number of interest cells in the *expected safe zone* that are viewed from the leaf position. Thus, we incorporate the expected information gain from each frontier as in Simmons et al. (2000). When the localization is good, the interest cells considered are the unexplored cells. When the localization is poor the interest cells considered are those marked as having low uncertainty. Fig. 9(d) shows how the expected unexplored cells from each leaf node has been added to the tree.

4.3.2. Evaluating the exploration tree

In order to evaluate the tree, we are going to give a value to each leaf node. These values are propagated back until the root node. We take into account the current degree of localization of the robot in the creation of the tree (following the hysteresis loop as explained in Section 4.1). In this sense, the tree can be constructed looking for frontiers or looking for past precise poses. This way, the interest cells counted for each leaf can be unexplored cells or past precise cells.

The value given to each leaf is directly proportional to the number of interest cells and inversely proportional to the cost of arriving to the position of the node. Thus, the value for each leaf

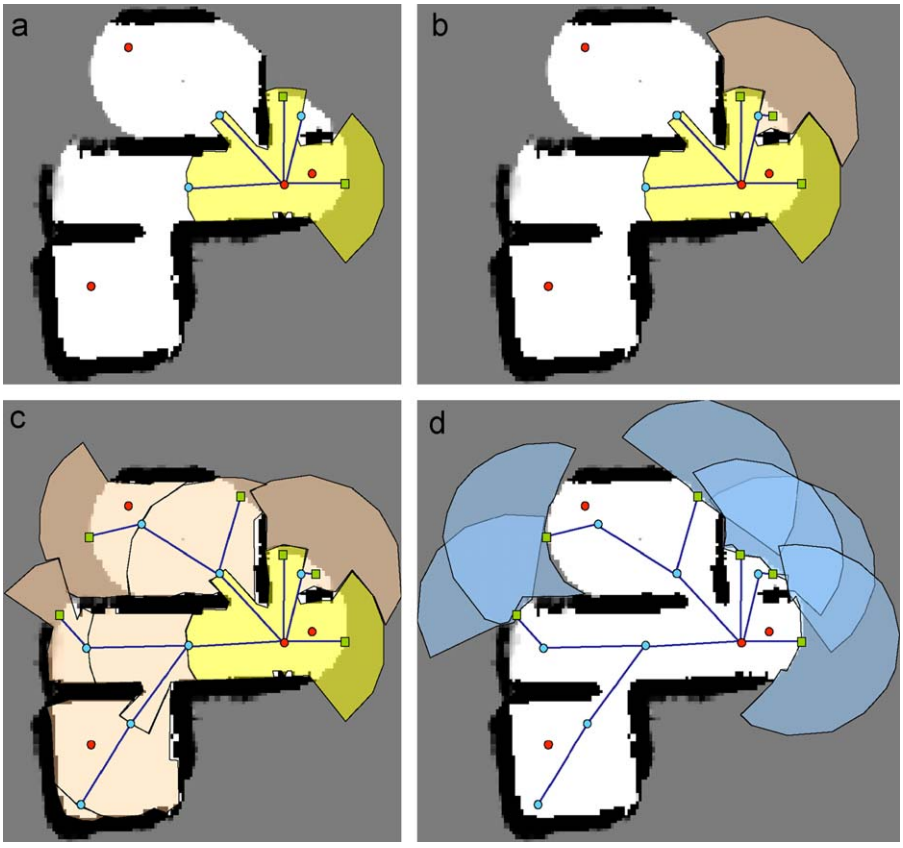

Fig. 9. The figure shows an example of creation of exploration tree. (a) The root node is added to the tree as well as the gateways (branches) and frontiers (leaves) found in the expected safe zone. (b) The closer branch is selected, a new associated zone is calculated and the new child nodes are added to the tree. (c) The process continue until no gateways are found. (d) The expected number of unexplored cells for each leaf node is counted.

Please cite this article as: Juliá, M., et al., A hybrid solution to the multi-robot integrated exploration problem. Engineering Applications of Artificial Intelligence (2010), doi:10.1016/j.engappai.2009.12.005

node is

$$V(n_L) = \frac{I_{n_L}}{C_{n_L}^2}, \quad (1)$$

where $V(n_L)$ is the value of the leaf node n_L, I_{n_L} is the number of interest cells in the node, and C_{n_L} is the cost to arrive to this node.

The back-propagation of the values for the tree is carried out by choosing the maximum between the values of the child nodes in each branch divided by the number of robots in the associated zone plus one. This way, the branches where other robots are present reduce their values. This is an advantage for a good coordination between the robots. Thus, the value for each branch node is

$$V(n_B) = \frac{\max_i V(n_B^i)}{B_{n_B}+1}, \quad (2)$$

being $V(n_B)$ the value of the branch node n_B, n_B^i the set of child nodes of n_B, and B_{n_B} the number of robots in the zone associated to the node.

The decision of the current state is made comparing the value for the first level nodes. Notice that these values are not affected when there are other robots in the current expected safe zone. This way, if two robots have very close positions, possibly they will have a similar tree with similar first level values. To achieve a better coordination we use a corrected value for the first level nodes:

$$V_c(n_R^i) = V(n_R^i)\prod_j d^2_{n_R^i, r^j}, \quad (3)$$

being $V_c(n_R^i)$ the corrected value for the set of child nodes of the root node n_R^i, $V(n_R^i)$ the normal value for the node n_R^i, and $d_{n_R^i, r^j}$ the distance between node n_R^i and robot r^j, where r^j is the set of robots in the current *expected safe zone*. Thus, the nodes that are far away from other robots increase their values.

Now, to make the final decision of the next state, we have to analyse the corrected value of the set of child nodes n_R^i of the root node (first level of the tree) and the current degree of localization of the robot. These are the three possible cases:

- *Explore current expected safe zone*: When localization is good and the node of maximum corrected value of n_R^i is a leaf.
- *Change zone*: When the node of maximum corrected value of n_R^i is a branch. We have to set the gateway of *go to gateway* behaviour to that node position.
- *Active localization*: When localization is poor and the node of maximum corrected value of n_R^i is a leaf.

Notice that the *change zone* state operates independently of the degree of localization. If the localization is good, the gateway selected leads the robot to frontiers, but when the localization is poor, the gateway selected leads the robot to past precise poses.

5. Experiments and results

5.1. Test bench

The method was tested in simulation in different scenarios whose appearance is shown in Fig. 10. Scenarios that represent hypothetical real places like *Scenario* 1 or *Scenario* 2 were chosen, whereas other scenarios such as, for example *Scenario* 3 or *Scenario* 4 are artificial. As shown, each scenario presents a different number of frontiers during the exploration because of its structure. For example, *Scenario* 3 is more prone to present a greater number of frontiers than *Scenario* 2 as it has more bifurcations. Furthermore, *Scenario* 4 presents narrow ways as the case explained in Section 4.2.2. In order to use the visual SLAM, a set of landmarks are situated randomly over the walls of each scenario. All the scenarios have fixed dimensions of 20×25 m.

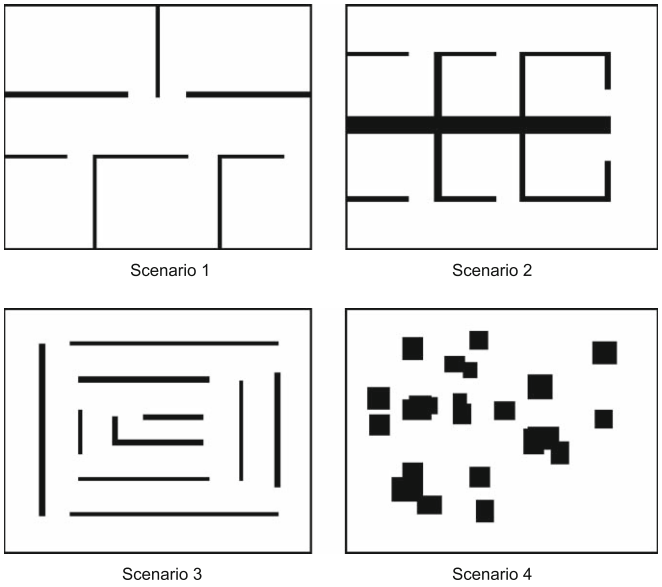

Fig. 10. The figure shows the four scenarios used to test the algorithm in simulation.

The test is made with a varying set of robots (changing from one to a group of eight robots). As each particle in the filter has to represent a possible path for all the robots at a time, the number of particles is increased proportionally to the number of robots. We start with 50 particles for one robot, we follow with 100 particles for two robots, until 400 particles for eight robots. To measure the degree of localization of each robot we make use of the standard deviation of the position of the robot σ^r:

$$\sigma^r = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(x_i^r - \overline{x}^r)^2 + (y_i^r - \overline{y}^r)^2}, \quad (4)$$

where M is the number of particles, (x_i^r, y_i^r) is the position of the robot r in particle i, and $(\overline{x}^r, \overline{y}^r)$ is the mean position of the robot with all the particles. This value σ^r is compared with the high and low thresholds. We have choose experimentally thresholds of $T_{low} = 0.1$ and $T_{high} = 0.2$.

The occupancy grid map is obtained with a resolution of 15 cm. The emulated sonar consists of a set of eight sensors with a maximum range of 5 m that cover the front of the robot. These sensors are situated in fixed intervals (with regard to the advance direction) from $-\pi/2$ to $\pi/2$ rad.

We assume that we are able to run the SLAM algorithm with a fixed time period of 3 s. We use this period as time unit. The reactive threads run synchronously with the SLAM. However, we increase the period for the deliberative threads in 5 *time units*. The exploration finishes when the final value of the exploration tree, evaluated by one of the robots, is zero. The linear speed of the robots is limited to 0.05 m/s and the angular speed is limited to 0.03 rad/s.

All the results for each scenario are given as the mean of several simulations changing the initial position of the robots. Next we explain the results regarding the speed of exploration, the map error and the state of the planner.

5.2. Speed results

The time of exploration is measured as the number of time units needed to complete the exploration. We can see these results in Fig. 11(a). As we can see, each scenario presents a different difficulty level and that is reflected in the exploration

Please cite this article as: Juliá, M., et al., A hybrid solution to the multi-robot integrated exploration problem. Engineering Applications of Artificial Intelligence (2010), doi:10.1016/j.engappai.2009.12.005

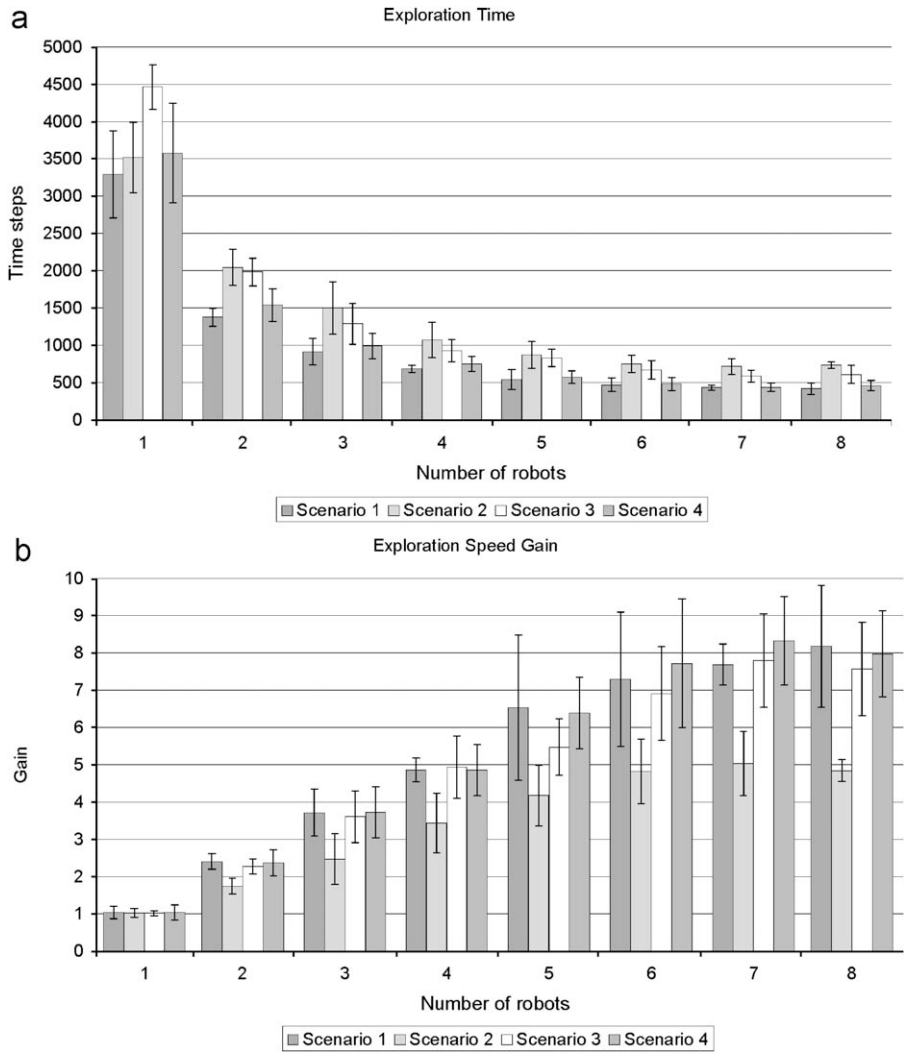

Fig. 11. Speed results. (a) Exploration time for the different scenarios and changing the number of robots. (b) Exploration speed gain because of the addition of robots to the group in relation to the exploration speed for a single robot.

time. As it was expected, the exploration time decreases gradually when the number of robots grows.

We can compare the time spent for the different number of robots with the time spent for a single robot. This way we can obtain the exploration speed gain that is introduced by the addition of robots. This gain is shown in Fig. 11(b). We can observe how the exploration speed gain is increased as the number of robots grows. With a number of 7 or 8 robots the gain begins to saturate. We can see that, regarding to the exploration speed gain, the different scenarios perform in different ways. The second scenario, which needs the robots to travel a long path, has very poor gain with the addition of robots. The other scenarios, which have more bifurcations, can exploit the robot coordination and they obtain a notable gain with the addition of robots.

5.3. Error results

The error is evaluated over the visual landmarks map. This is carried out by comparing the position of each landmark in the visual landmark map associated to the most probable particle with their real positions. The resulting error will be expressed as the root mean square. The error results of the simulation are shown in Fig. 12.

The error in the visual landmark map obtained is small in relation with the dimensions of the explored zone. The error tends to decrease as the number of robots increases. However, there is not a clear dependence. The map error depends on a lot of factors. On the one hand, as the number of robots grows, more observations are added to the system and the robot has to cover a minor area travelling a shorter path. Thus, the results should be better. On the other hand, despite the increasing number of particles in the filter proportionally to the number of robots, each particle is a worst representation of the state of the robot because they have to represent the position for all the robots in the system. Furthermore, the error depends on the path performed by the robots, but the system includes the active localization state in order to return to previously explored zones when the uncertainty in the position of the robots is high. As a result, we cannot observe a clear dependence of the error with the number of robots, only we can see a small tendency to go down.

5.4. State of the planner results

In order to check the functionality of the planner, we will analyse the states selected. The results of the simulation are shown in Fig. 13. This figure shows the percentage of time that the robots are working in each state. We have divided the *change zone* state in

Please cite this article as: Juliá, M., et al., A hybrid solution to the multi-robot integrated exploration problem. Engineering Applications of Artificial Intelligence (2010), doi:10.1016/j.engappai.2009.12.005

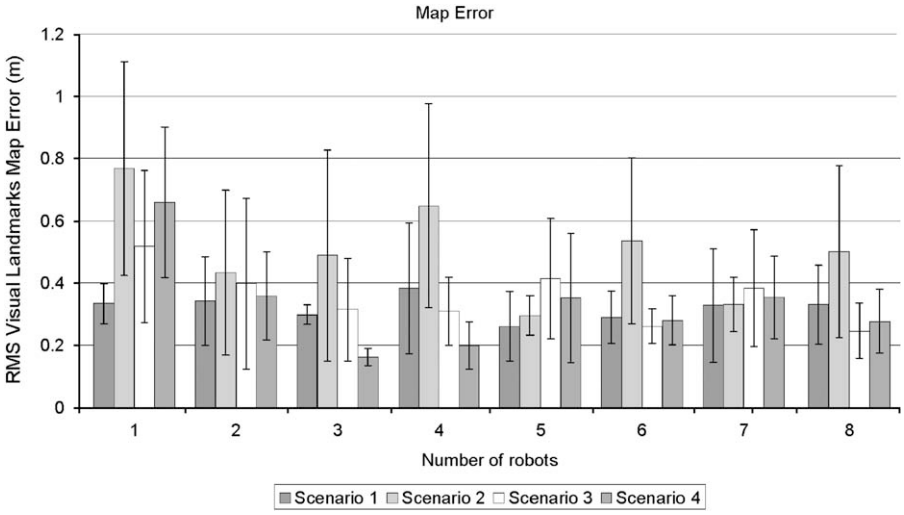

Fig. 12. The figure shows the root mean square error committed in the map of visual landmarks for the different scenarios and changing the number of robots.

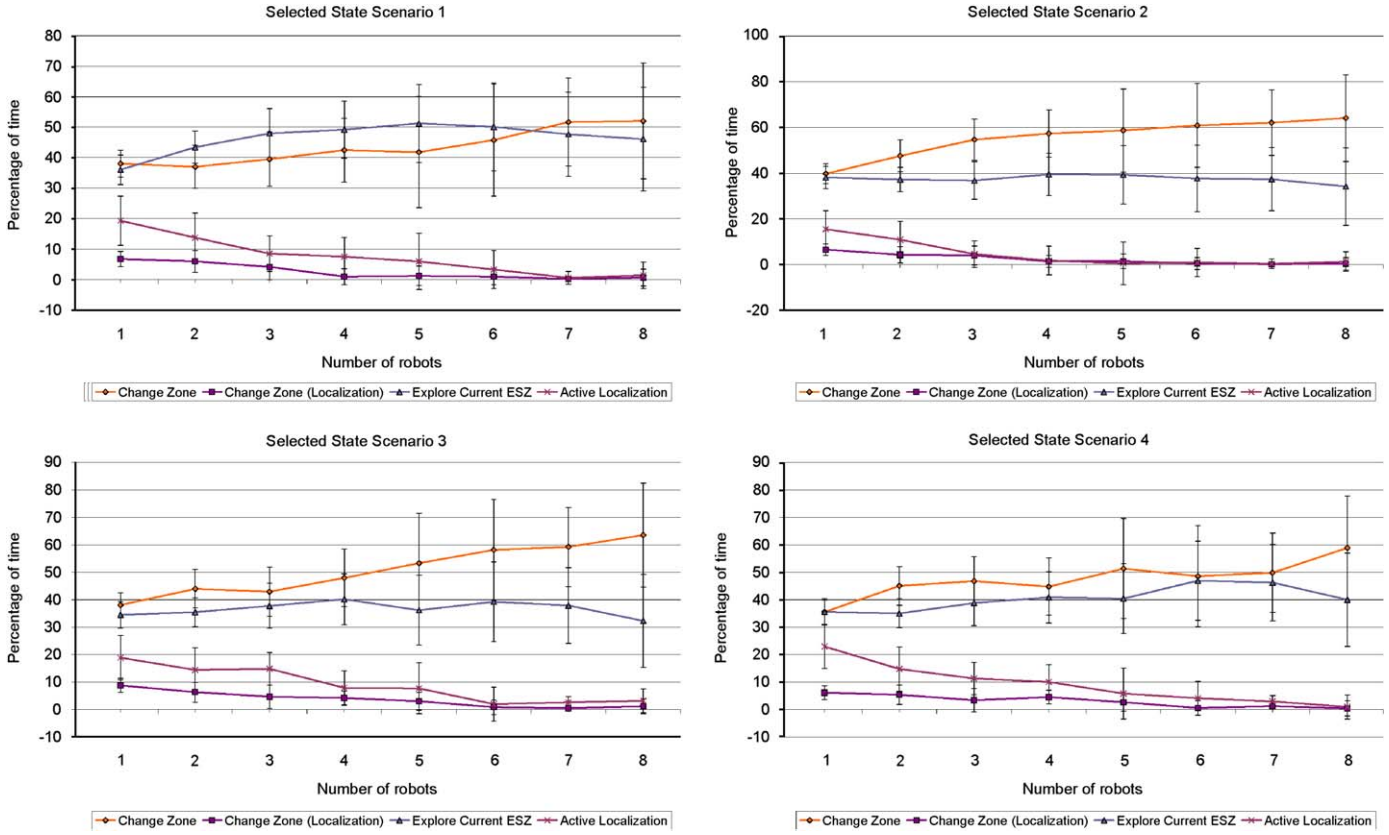

Fig. 13. The figure shows the mean percentage of time that the robots are running in each state for the different scenarios and changing the number of robots.

two cases depending on whether the selected gateway is leading the robot to frontiers or to past precise poses for localization.

As we can see, the two exploring states, *explore current expected safe zone* and *change zone*, are the main states and both have the same importance level. As we said, reactivity and deliberation work together. We can observe that as the number of robots increases the percentage of time in the *change zone* state increases. This is because of the fact that as the number of robots grows, more coordination is needed.

The two localization states, *active localization* and *change zone* when the gateway leads to past poses, require an small time. We can observe that this time decreases when the number of robot grows.

Fig. 14 shows the average number of state transitions per robot. It is reduced as the number of robots grow. Notice that, as the number of robots grow, the paths performed by each robot are shorter and the localization is better. These facts produce this reduction in the number of transitions for a large number of robots.

5.5. Performance comparison with classic methods

Fig. 15 shows a comparison of the performance of our technique in relation to a classic path planning exploration where each robot navigates always towards its nearest frontier (Yamauchi, 1998). The values of Fig. 15 correspond to

Please cite this article as: Juliá, M., et al., A hybrid solution to the multi-robot integrated exploration problem. Engineering Applications of Artificial Intelligence (2010), doi:10.1016/j.engappai.2009.12.005

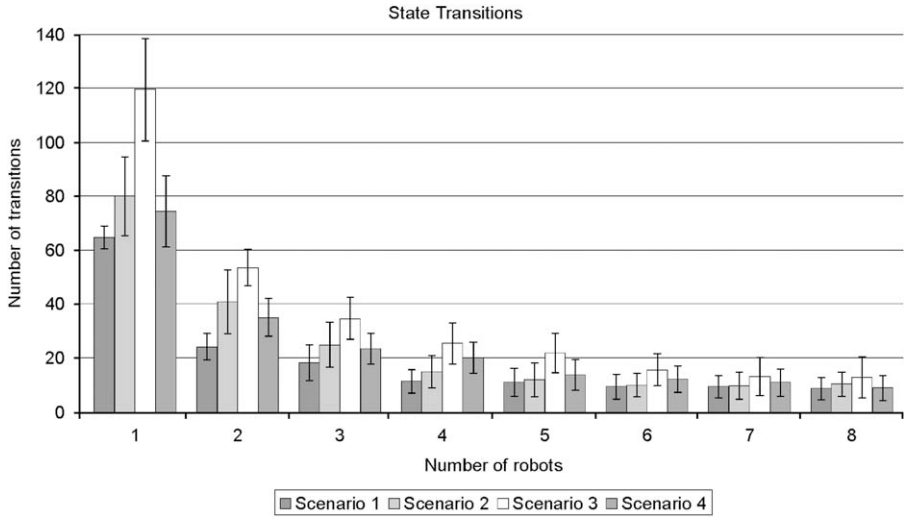

Fig. 14. The figure shows the average number of state transitions for the different scenarios and changing the number of robots.

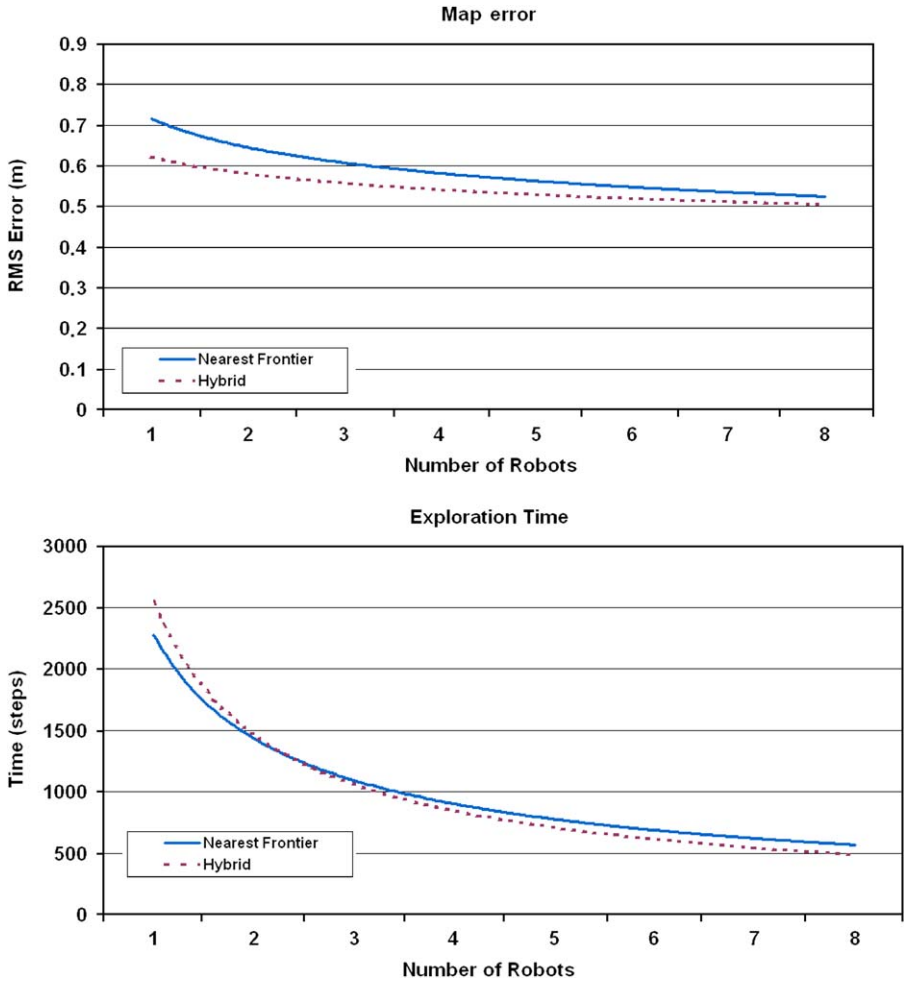

Fig. 15. Performance comparison between the hybrid method and the nearest frontier path planning technique.

experiments in *Scenario* 1. As we can see, the error is always higher using the classic path planning to the nearest frontier. The exploration time is higher with out technique when having a small team of robots since they use some time returning to previously explored zones. However, when there are enough robots, as our technique have a better coordination, the exploration time is reduced below the time used by the path planning approach to the nearest frontier.

Please cite this article as: Juliá, M., et al., A hybrid solution to the multi-robot integrated exploration problem. Engineering Applications of Artificial Intelligence (2010), doi:10.1016/j.engappai.2009.12.005

6. Comparison with other integrated exploration techniques

6.1. Juliá et al.'s approach

Juliá et al. (2008) use an approach consisting of several compositions of basic behaviours that are sequenced with a finite state automata. These macro-behaviours are *exploration*, *active localization*, and *escape from local minima*. This technique is limited by the way the local minima affects the speed of the exploration. The low level behaviours are similar to the exposed in this paper since this is an evolution of that approach solving the problem of local minima in a more efficient way. Furthermore, the coordination is improved. The Juliá et al. (2008) approach only have the coordination due to the *avoid obstacles behaviour*. In this sense, the coordination was poor in some scenarios. However, in this paper the introduction of the *expected safe zone* and the consideration of the topology of the environment using the *exploration tree* in the deliberative layer improves the coordination notably. The behaviour used to return to previously explored zones was an attraction to precise landmarks, however, in our new approach this is carried out by means of navigation to past poses where the robot was well localized. In this way our method does not require a landmark-based SLAM.

6.2. Makarenko et al.'s approach

Makarenko et al. (2002) use a cost-utility path planning approach. Their utility function considers the information gain of the occupation grid map at the target point as well as the localizability of the robot at this point. In this way, the frontiers that are near to landmarks have a higher utility. They consider only the frontiers as possible targets, however, they suggest using locations where the chances of relocalizating are high when the covariance of the pose of the robots is too large. In this sense, we have implemented a solution that is similar to the one that they suggest by means of leading the robots to past poses. However, we do not distinguish between frontiers in function of the localizability. Including this kind of distinction in our behaviour based system is a future line of work. In contrast with our method, Makarenko et al. (2002) is a mono-robot approach that does not provide coordination for several robots.

6.3. Stachniss et al.'s approach

Stachniss et al. (2005a) is one of the most relevant integrated exploration techniques. It is also based on the information gain, which may come from the incorporation of new zones to the map or from reducing the uncertainty of the map. Following this objective of information gain, it analyses three types of possible destinations: frontiers, past poses, and loop closing points. The loop closing points are obtained from a topological map (Stachniss et al., 2005b). The expected information gain is integrated for the full planned path until reaching each possible target. This information gain is analysed in connection to the Rao Blackwellized Particle Fixlter with occupancy grid maps and laser scan matching that it uses for SLAM. Stachniss et al. (2005a) does not consider a multi-robot approach.

6.4. Freda et al.'s approach

Freda et al. (2006) use a sensor-based random tree (SRT). The tree is expanded as new candidate destinations near the frontiers of the sensor coverage are selected. These candidate destinations are evaluated considering the reliability of the expected observable features from that points. The tree is used to navigate back to past nodes with frontiers when no frontiers are present in the current sensor coverage. Furthermore, this method includes a homing operation in order to close the loop at the end of the exploration. However, it does not consider intermediate loop closing or returns to past positions to improve the localization. The authors explain in other paper how to coordinate several robots using SRT (Franchi et al., 2007).

6.5. Tovar et al.'s approach

Tovar et al. (2006) considers only random points near the frontiers as potential targets. A utility function evaluates that points considering the utility for exploration and localization of the full path until reaching that point. A potential target with a high number of landmarks observable while navigating to that point obtains a high utility. The order of navigation between potential targets is evaluated with a decision tree that considers the different utilities. This technique incorporates also multi-robot coordination by means of introducing the possible routes for each robot in the decision tree. From the point of view of coordination this is a good option but computationally expensive, highly centralized and poorly scalable. Moreover, it does not consider explicitly returning to past poses.

7. Conclusions and future works

A hybrid reactive/deliberative approach to the multi-robot integrated exploration problem has been developed. The design of the reactive and deliberative processes is fully oriented to the exploration, having both the same importance level. The reactive system is based on the potential field generated by several basic behaviours. As stated by many authors, potential field based methods present the disadvantage of local minima. However, restricting the model to the *expected safe zone* of the robot, as introduced in this paper, avoids the presence of local minima. Simultaneously, a planner builds up a decision tree in order to decide between exploring the current *expected safe zone* or travelling to another zone by means of navigating to a *gateway cell*, that has been also defined in this paper. The algorithm that the planner uses to create and evaluate the *exploration decision tree* has been explained. Besides, this algorithm considers also the uncertainty in the location of the robots in order to return to previously explored places when the uncertainty becomes significant. This fact improves the quality of the resulting map. Several simulations have been presented that demonstrate the validity of the approach.

As future works, we consider the extension of the approach in real dynamic environments, adding some techniques to learn automatically the multiple settings of the system. We will develop a full distributed system by separating the SLAM process between the robots. Inter-robot communication and mechanisms for meeting the robots to align their maps will be added. Semi-operated models that integrate the commands expressed by a human operator in the exploration task will also be studied.

Acknowledgement

This work has been supported by the Spanish Government (Ministry of Science and Innovation). Project: 'Sistemas de percepción visual móvil y cooperativo como soporte para la realización de tareas con redes de robots'. Ref.: DPI2007-61197.

References

Amigoni, F., 2008. Experimental evaluation of some exploration strategies for mobile robots. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'08), Pasadena, CA, USA.

Arkin, R., Diaz, J., 2002. Line-of-sight constrained exploration for reactive multiagent robotic teams. In: Proceedings of the International Workshop on Advanced Motion Control (AMC'02), Maribor, Slovenia.

Ballesta, M., Gil, A., Reinoso, O., Martínez-Mozos, O., 2008a. Evaluation of interest point detectors for visual slam. International Journal of Factory Automation, Robotics and Soft Computing 4, 86–95.

Ballesta, M., Reinoso, O., Gil, A., Juliá, M., Payá, L., 2008b. Analysis of map alignment techniques in visual slam systems. In: Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation (EFTA'08), Hamburg, Germany.

Burgard, W., Moors, M., Stachniss, C., Schneider, F., 2005. Coordinated multi-robot exploration. IEEE Transactions on Robotics 21 (3), 376–386.

Cao, Y., Fukunaga, A.S., Kahng, A.B., 1997. Cooperative mobile robotics: antecedents and directions. Autonomous Robots 4 (1), 7–27.

Chao-xia, S., Bing-rong, H., Yan-qing, W., 2008. Cooperative exploration by multi-robots without global localization. International Journal of Advanced Robotic Systems 5 (2), 129–138.

Farinelli, A., Iocchi, L., Nardi, D., 2004. Multi-robot systems: a classification focused on coordination. IEEE Transactions on Systems, Man and Cybernetics B 34 (5), 2015–2028.

Fernandez, J.A., Gonzalez, J., Mandow, L., de la Cruz, J.L.P., 1999. Mobile robot path planning: a multicriteria approach. Engineering Applications of Artificial Intelligence 12 (4), 543–554.

Fox, D., Ko, J., Limketkai, B., Schulz, D., Stewart, B., 2006. Distributed multi-robot exploration and mapping. Proceedings of the IEEE 94 (7), 1325–1339.

Franchi, A., Freda, L., Oriolo, G., Vendittelli, M., 2007. A randomized strategy for cooperative robot exploration. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'07), Roma, Italy.

Freda, L., Loiudice, F., Oriolo, G., 2006. A randomized method for integrated exploration. In: Proceedings of the IEEE-RSJ International Conference on Intelligent Robots and Systems (IROS'06), Beijing, China.

Garrido, S., Moreno, L., Blanco, D., 2008. Exploration of a cluttered environment using Voronoi transform and fast marching. Robotics and Autonomous Systems 56 (12), 1069–1081.

Gil, A., Reinoso, O., Ballesta, M., Juliá, M., 2010. Multi-robot visual slam using a Rao–Blackwellized particle filter. Robotics and Autonomous Systems 58 (1), 68–80.

Gonzalez-Baños, H.H., Latombe, J.C., 2002. Navigation strategies for exploring indoor environments. International Journal of Robotics Research 21 (10).

Juliá, M., Gil, A., Payá, L., Reinoso, O., 2008. Local minima detection in potential field based cooperative multi-robot exploration. International Journal of Factory Automation, Robotics and Soft Computing 3.

Juliá, M., Gil, A., Payá, L., Reinoso, O., 2008. Potential field based integrated exploration for multi-robot teams. In: Proceedings of the International Conference on Informatics in Control, Automatics and Robotics (ICINCO'08), Funchal (Madeira), Portugal.

Lau, H., 2003. Behavioural approach for multi-robot exploration. In: Proceedings of the Australasian Conference on Robotics and Automation (ACRA'03), Brisbane, Australia.

Makarenko, A., Williams, S., Bourgoult, F., Durrant-Whyte, F., 2002. An experiment in integrated exploration. In: Proceedings of the IEEE-RSJ International Conference on Intelligent Robots and Systems (IROS'02), Lausanne, Switzerland.

Montemerlo, M., Thrun, S., 2003. Simultaneous localization and mapping with unknown data association using fastslam. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'03), Taipei, Taiwan.

Moravec, H., Elfes, A., 1985. High resolution maps from wide angle sonar. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'85).

Murphy, L., Newman, P., 2008. Using incomplete online metric maps for topological exploration with the gap navigation tree. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'08), Pasadena, CA, USA.

Newman, P., Bosse, M., Leonard, J., 2003. Autonomous feature-based exploration. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'03), Taipei, Taiwan.

Posadas, J.L., Poza, J.L., Simó, J., Benet, G., Blanes, F., 2008. Agent-based distributed architecture for mobile robot control. Engineering Applications of Artificial Intelligence 21 (6), 805–823.

Prestes, E., Engel, P.M., Trevisan, M., Idiart, M.A., 2002. Exploration method using harmonic functions. Robotics and Autonomous Systems 40 (1), 25–42.

Rekleitis, I., Dudeck, G., Milios, E., 1997. Multi-robot exploration of an unknown environment, efficiently reducing the odometry error. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'97), Nagoya, Japan.

Rekleitis, I., Sim, R., Dudek, G., Milios, E., 2001. Collaborative exploration for the construction of visual maps. In: Proceedings of the IEEE-RSJ International Conference on Intelligent Robots and Systems (IROS'01), Wailea, Hawaii.

Rocha, R., Ferreira, F., Dias, J., 2008. Multi-robot complete exploration using hill climbing and topological recovery. In: Proceedings of the IEEE-RSJ International Conference on Intelligent Robots and Systems (IROS'08), Nice, France.

Santosh, D., Achar, S., Jawahar, C., 2008. Autonomous image-based exploration for mobile robot navigation. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'08), Pasadena, CA, USA.

Schmidt, D., Luksch, T., Wettach, J., Berns, K., 2006. Autonomous behavior-based exploration of office environments. In: Proceedings of the International Conference on Informatics in Control, Automation and Robotics (ICINCO'06), Setúbal, Portugal.

Simmons, R., Apfelbaum, D., Burgard, W., Fox, D., Moors, M., Thrun, S., Younes, H., 2000. Coordination for multi-robot exploration and mapping. In: Proceedings of the AAAI National Conference on Artificial Intelligence, Austin, TX, USA.

Stachniss, C., Grisetti, G., Burgard, W., 2005a. Information gain-based exploration using Rao–Blackwellized particle filters. In: Proceedings of Robotics: Science and Systems (RSS'05), Cambridge, MA, USA.

Stachniss, C., Haehnel, D., Burgard, W., Grisetti, G., 2005b. Actively closing loops in grid-based fastslam information. RSJ Advanced Robotics 19 (10), 1059–1080.

Stachniss, C., Mozos, O.M., Burgard, W., 2006. Speeding-up multi-robot exploration by considering semantic place information. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'06), Orlando, FL, USA.

Tovar, B., Muñoz-Gómez, L., Murrieta-Cid, R., Alencastre-Miranda, M., Monroy, R., Hutchinson, S., 2006. Planning exploration strategies for simultaneous localization and mapping. Robotics and Autonomous Systems 54 (4), 314–331.

Wullschleger, F.H., Arras, K.O., Vestli, S.J., 1999. A flexible exploration framework for map building. In: Third European Workshop on Advanced Mobile Robots (Eurobot '99).

Wurm, K.M., Stachniss, C., Burgard, W., 2008. Coordinated multi-robot exploration using a segmentation of the environment. In: Proceedings of the IEEE-RSJ International Conference on Intelligent Robots and Systems (IROS'08), Nice, France.

Xiaoping, Y., Ko-Cheng, T., 1997. A wall-following method for escaping local minima in potential field based motion planning. In: Proceedings of the International Conference on Advanced Robotics (ICAR'97), Monterey, CA.

Yamauchi, B., 1997. A frontier based approach for autonomous exploration. In: Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA'97), Monterey, CA, USA.

Yamauchi, B., 1998. Frontier-based exploration using multiple robots. In: Proceedings of the International Conference on Autonomous Agents, Minneapolis/Saint Paul, Minnesota, USA.

Zlot, R., Stentz, A., Dias, M.B., Thayer, S., 2002. Multi-robot exploration controlled by a market economy. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'02), Washington, DC, USA.

Please cite this article as: Juliá, M., et al., A hybrid solution to the multi-robot integrated exploration problem. Engineering Applications of Artificial Intelligence (2010), doi:10.1016/j.engappai.2009.12.005