International Journal of Innovative Computing, Information and Control
Volume **-**, Number **0-**, **- 2010**

ICIC International ©2010 ISSN 1349-4198
pp. **1–20**

BEHAVIOUR BASED MULTI-ROBOT INTEGRATED EXPLORATION

Miguel Juliá, Oscar Reinoso, Arturo Gil, Mónica Ballesta and Luis Payá

Industrial Systems Engineering Department
Miguel Hernandez University
Avda. Universidad s/n. Edif. Quorum V, 03202 Elche-Alicante, Spain
{mjulia, o.reinoso, arturo.gil, m.ballesta, lpaya}@umh.es

Received April 2010

Abstract. *This paper presents an approach to the integrated exploration problem for a team of mobile robots. This technique is based on a combination of several basic behaviours that model a potential field. These behaviours are designed to quickly explore the environment jointly with a visual SLAM technique. As a novelty, this method considers returning to previously explored areas when the localization uncertainty is high. Consequently, the accuracy obtained in the construction of the maps is higher than with other classical exploration techniques. The known problem of local minima in potential field based techniques is also considered. In this sense, a strategy of detection and escape from local minima is used. Several simulations show the validity of the approach.*

Keywords: integrated exploration, cooperative robots, potential fields, behaviour based robotics

1. **INTRODUCTION.** During the last years, applications that require the deployment of mobile robots have become more frequent. These approaches require the navigation through unstructured and unknown environments in which the task of exploration is crucial. Exploration consists in the coverage of an unknown environment by a robot or a group of mobile robots building a common map at the same time. The use of a team of robots is an advantage [1], since the exploration time can be reduced and the precision of the maps can be improved [2]. Exploration techniques can be applied to surveillance, search and rescue services, map building or planetary exploration.

The problem of exploration is related to the Simultaneous Localization and Mapping (SLAM). The maps built by a robot while it is exploring the environment can consist of occupancy grids from the information supplied by range sensors or they can consist of visual features from the environment [3]. Figure 1 graphically shows the concepts of exploration and SLAM and how they are related [4, 5]. On the one hand, SLAM techniques are able to build a map and locate the robots within it, nevertheless they are passive due to the fact that they do not control the motion of the robots. On the other hand, classic exploration algorithms direct the robots trying to perform a fast coverage of the environment. However, the SLAM algorithms are affected by the performed trajectories. That means that the results in terms of accuracy depend on the trajectories followed by the robots [4]. Even though path generation and following has been studied as the case of parallel robots [6], the trajectories performed by the robots when they explore a completely undetermined environment play an essential role. In this sense, when the robots travel through unknown environments the localization uncertainty continuously grows. This fact may lead to inaccurate or useless maps. Therefore, in order to obtain a precise map it is necessary that the robots make also movements considering the uncertainty in their localization. As Figure 1 shows, the exploration algorithms that take into account all

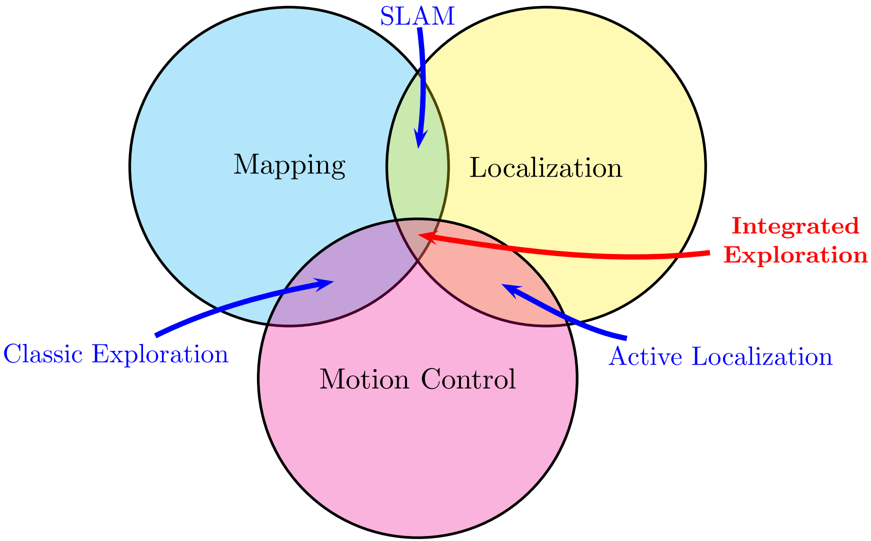


Figure 1. The figure shows the concept of integrated exploration and its relation with the SLAM.

these aspects of the SLAM are commonly called Integrated Exploration Algorithms [5]. This paper is focused on the Integrated Exploration problem for a team of mobile robots.

The objective of this approach is to decide appropriate control actions for each robot of the team in order to map the desired area with full coverage. We propose in this paper a method to resolve this problem using a coordinated and integrated model. The planning of the control actions coordinates the movements of the robots in such a way that they spread over the environment performing a fast coverage. At the same time the motion control is integrated with the SLAM algorithm by means of considering the uncertainty in the position of the robots. The robots try to keep their uncertainty below a certain value so that a bad localization does not cause an inaccurate and completely worthless map. In order to reduce the uncertainty, the algorithm is designed to make the robots return to previously explored zones when the localization uncertainty is too high. By returning to previously explored areas the uncertainty is reduced. Thus, we improve the quality of the built map.

Some authors only perform a classic exploration, that is moving the robots while mapping the environment without improving the odometric localization [7]. Other techniques include classic exploration simultaneously with SLAM techniques [8]. However, in contrast to integrated exploration approaches [9, 5], they do not consider the uncertainty of the SLAM when selecting the movements of the robots. Other authors have focused its research in coordinating the team of robots [10, 11]. However, very few papers have discussed a cooperative and integrated exploration in the same approach [12]. The main contribution of this paper is that we have implemented it using relative low cost sensors (only sonar and a stereo camera) and in contrast to [12] that uses a complex and computationally expensive deliberative planning our technique is implemented as a simple reactive behaviour based system.

The remainder of this paper is structured as follows. Section 2 presents the state of the art in the field of exploration. In Section 3 the proposed approach is explained in detail. Section 4 presents the experiments that were carried out to test the method and their results. Section 5 analyses the performance of the technique and compares it with other techniques. The main conclusions and future works are exposed in Section 6.

2. **RELATED WORK.** The great majority of the exploration techniques use the frontier concept introduced by Yamauchi [13]. Using an occupancy grid map, frontiers can be defined as free cells next to an unknown cell. Since these cells lead to the unknown areas of the environment, which are the objective of the exploration, a good strategy consist in planning paths that lead the robot to frontier cells.

When using a team of robots, the problem is extended to how to coordinate them to reduce the exploration time or improve the quality of the map. In that sense, either a centralized or a distributed approach [14] could be used. Furthermore, considering the planning level we can classify the exploration techniques in two groups:

- Deliberative: The exploration is directed by a high level layer of the architecture that evaluates a long term motion planning with full knowledge of the global estimated map and positions of the robots. A low level reactive layer may be used to avoid dynamic obstacles.
- Reactive: The exploration is led by low level behaviours that work in real-time with a partial knowledge of the map and evaluating only a short term movement.

In the first group, the usual operation mode consist in planning a path to a frontier cell. These techniques differ in the way they coordinate the robots, assigning them different frontiers. In [7, 8] the nearest frontier is used. It is also possible to use a cost-utility model where the cost is the length of the path to achieve a frontier and the utility can be related to the expected information gain from the frontier [15, 16], or a function of the frontiers assigned to other robots [10]. An agents model, where the robots negotiate the assigned frontiers [11], can also be used. Another way to select the next goal can be using a Sensor-Based Random Tree [17] where the environment is explored in a depth-first search manner.

In the second group, the exploration is carried out with basic reactive behaviours that are usually modelled as potential fields [18]. In [19] the behaviours: *Probe*(Go to free space) and *Avoid Past* (Rejection from previously explored zones) are used in addition to common behaviours as a repulsive *Avoid obstacles* behaviour. In [20] the frontier concept is used through a *Move to frontier* behaviour and an *Avoid Other Robots* behaviour that spreads the robots over the environment. However, using this kind of behaviours, local minima are likely to appear. A local minimum takes place when the behaviours cancel each other in such a way that a point of minimum potential appears. Since the robots travel following the zones of lower potential they may be blocked in that points. A common solution to this problem is to plan a path to the nearest frontier [21]. Another solution to this problem but more inefficient is using a wall following strategy [22]. Harmonic functions have also been used for designing systems without local minima [23]. However, they need to update a global potential field, so it may present low scalability and a high computational cost. Furthermore, the potential fields generated with harmonic functions can not be added preserving their properties. Therefore, it is difficult to integrate this technique with a multi-objective behaviour.

As we explained in Section 1, classical exploration approaches always direct the robots trying to maximize the information gain but they do not take into account the quality of their localization. Since the accuracy in the SLAM problem is affected by the trajectories of the robots, it is necessary that the robots make also movements considering the uncertainty in their localization. As we said, this kind of techniques are called Integrated Exploration Algorithms. In the field of integrated exploration, several techniques have been applied. In [4] the uncertainty is reduced by actively closing loops with previously explored areas using a topological map. In [24] the certainty over the pose of the robots during an exploration that uses a parametric curve trajectory is recovered. This is carried out by means of modifying the parameters in order to return to previously explored zones. In [5] the uncertainty in the localization is included as a part of a cost-utility model in the assignment of destinations to robots. In [25], using a sensor based tree, the candidates points to expand the tree that are situated near precise landmarks have higher priority. A complex utility function that considers the number of landmarks that are observable in a path to potential targets near the frontiers is used in [12]. 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.

Other authors have focused in other problems that appear when using a team of robots. For instance, [26] considers the case of a constrained communication network range and introduces some role changing to regroup the robots and restore the communication. In [14], the initial positions of the robots are considered as unknown. In this case each robot has to built its own map. When two robots meet they align and fuse their maps. Other authors deal with non-Markovian sequential task as for instance in [27] where a recurrent neural network is used.

3. **APPROACH.**

Figure 2 shows the structure of the architecture used in this approach. As it can be seen, our technique consists of the set of robots and a central unit running a multi-robot SLAM algorithm. The robots consist of a differentially driven platform equipped with a stereo camera, 8 sonar sensors, that cover 180 degrees in front of the robot, and a odometry sensor. The readings of the odometry, sonar and the 3D positions of the landmarks extracted from the images captured with the stereo camera of each robot are sent to the central unit through a local network. At the central unit a centralized SLAM process builds a set of maps and localizes the robots in them. As the maps are built cooperatively, each robot benefits from the information of the other robots. These maps and the localization information are broadcast. Thus, in the system each robot knows the maps and the position of all the robots of the team. Each robot runs a reactive behaviour-based motion control algorithm to obtain the appropriate movement commands. We have chosen a reactive approach because it allows to integrate multiple objectives simultaneously as simple behaviours. First of all, different sets of behaviours are activated depending on the state of a finite state automata (FSA). Then, the active behaviours are combined obtaining a desired new direction. With the purpose of avoiding local minima that could be produced from that combination of behaviours, a local minima detector activates an auxiliary path planning module to escape from local minima. Finally, a controller sends appropriate control actions to move the robot in the desired direction.

3.1. **SLAM.**

As we previously mentioned, each robot is equipped with a stereo vision system. With this sensor, we can identify visual feature points, obtain pairs of matching features between the images and determine its 3D position in the space with a good precision. For this reason, the process of localization of the robots is performed with the visual landmarks information extracted from the images of the cameras. A SLAM algorithm consisting in a Rao-Blackwellized particle filter that creates a visual landmark map and returns the positions of the robots is used for this purpose. This SLAM technique is centralized using the data of all the robots [28]. Furthermore, in order to navigate avoiding obstacles we need also an occupancy map. Using the localization returned by the visual SLAM and the sonar readings an occupancy grid mapping technique is applied. Finally, we also use an auxiliary map consisting in a binary grid to save positions where we can localize the robots with a good precision. So three different maps are employed in the proposed integrated exploration algorithm: a visual landmarks map, an occupancy gridmap and finally a past precise poses map.

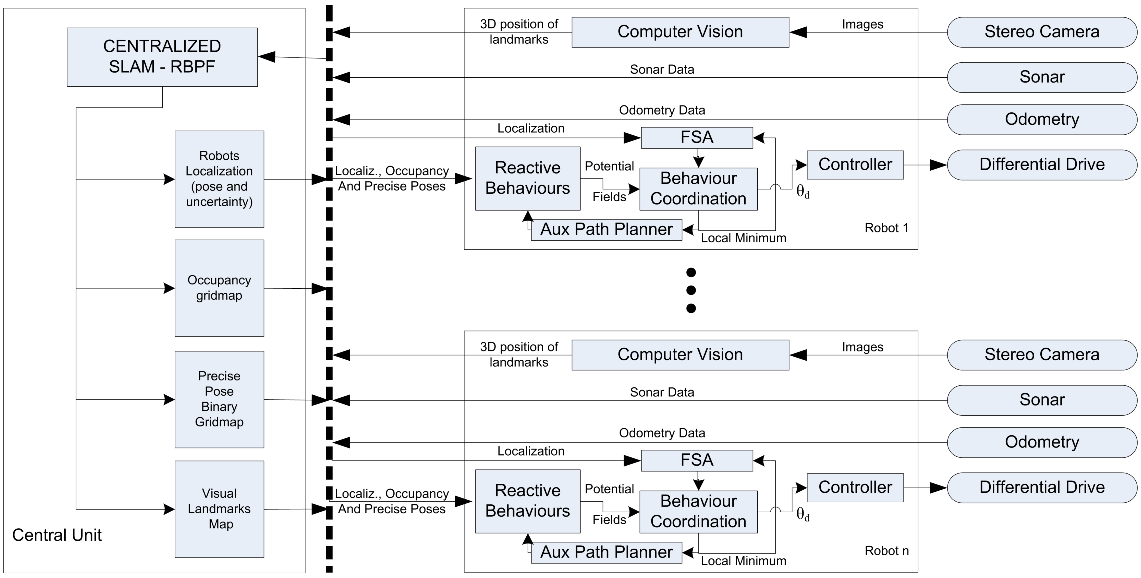


FIGURE 2. Architecture

3.1.1. *Visual Landmarks Map.* Landmark-based maps are used to represent the environment. This kind of map consists of a cloud of 3D points corresponding to the estimated positions of the detected landmarks along with its covariance. The visual landmarks can be extracted using stereo cameras with appropriate feature detectors. A visual descriptor can help during the matching process. For instance in [29], after an analysis of feature detectors and visual descriptors, the Harris corner detector and the SURF descriptor are recommended. In our simulations we use a set of predetermined landmarks randomly positioned over the obstacles of the virtual environment with a fixed descriptor. We assume that the robots are able to detect landmarks in a range of $8m$. We take into account that the obstacles in the environment may block the visibility of the landmarks.

In this paper, we carry out the SLAM by means of a Rao-Blackwellized particle filter using the model exposed in [28]. The approach was based on the FastSLAM algorithm [30] but using visual landmarks and extended to multiple robots. As this SLAM algorithm is processed jointly in a centralized way, the robots take advantage of the data gathered by the other robots to achieve a better localization. The basic idea of the FastSLAM algorithm is that the SLAM problem can be decomposed into two parts: the estimation of the paths followed by the robots, which is represented by a set of particles that evolves with the control actions given to the robots, and the estimation of a landmarks map associated to each one of the possible paths using an extended Kalman Filter for each landmark. This way, we have a visual landmark map for each particle. The particles are weighted according to the innovation of the new observations and subsequently they are resampled. The full explanation of this technique can be found in [28]. The results of this SLAM algorithm are the corresponding to the most probable particle, that is, the most probable positions of each robot and the most probable visual landmark map with their uncertainty.

3.1.2. *Occupancy Gridmap.* The visual landmarks map does not represent the occupation of the environment, that is, whether an area can be safely traversed by the robots or not. In order to move the robots safely we need this information. For this reason we use also an occupancy grid map that represents the occupation probability of the space [31, 32]. To build this map we use the measurements of a sonar with a range of $5m$ and the approach of Moravec and Elfes [31]. This algorithm builds a map considering the positions of the

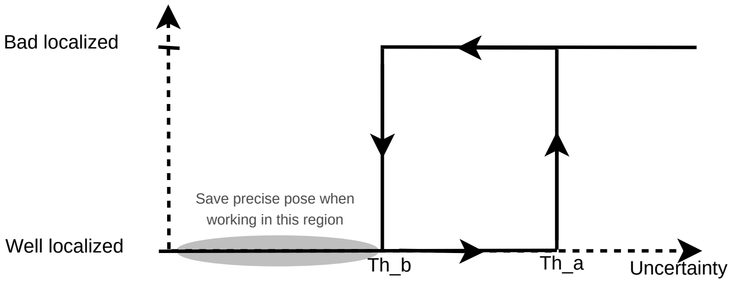


Figure 3. Localization hysteresis model

robots given by the visual SLAM algorithm. We run the algorithm for each particle of the filter having a different occupation map per each particle. We use the map corresponding to the most probable particle for the motion control.

Depending on the occupation probability for each cell, we can classify the cells as free (less than 50% of occupancy probability), occupied (more than 50%) or unknown (50%). With this information, frontier cells are defined as the free cells that are contiguous to one or more unknown cells. As this map is also broadcast to all the robots, they know the frontiers generated by the actions of the other robots. In this manner, the robots know which zones have already been visited by other robots and they can determine appropriate actions in order to contribute to the team exploration task.

3.1.3. *Past precise poses map.* When the robots travel through unknown environments their localization uncertainty grows. Sometimes, when the localization of a robot is too poor we will need to guide the robot to previously explored zones in order to reduce its uncertainty. This way, provided that the robots are the most of the time well localized the accuracy of the map improves.

We can measure the quality of the localization of the robot by means of evaluating the standard deviation of its position in the particle filter (σ^r):

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

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.

Following a hysteresis model as shown in Figure 3, we have experimentally fixed two localization thresholds (Th_a, Th_b) that determine when a robot is considered to have a good or a poor localization. In order to reduce the uncertainty in the localization of the robots, we need to save positions where we know that the robot is able to reduce it. Therefore, we save in a binary grid map the past positions where the robot position had a standard deviation below the low threshold Th_b. The other threshold (Th_a) indicates when the robot is too bad localized and we have to initiate the action of going to the past precise poses stored in this map to recover a good localization. This process concludes when the robot reduces its uncertainty below Th_b and is considered again as well localized. Note that, as all maps are shared by all the robots, the robots can go to past poses of the other robots to reduce the uncertainty in their positions.

3.2. **Design of Behaviours.** Our approach to the exploration problem consists of several basic behaviours. Each behaviour considers one aspect of the exploration and evaluates a motion pattern defined by a potential field. These potential fields are generated by simple discrete gaussian functions, which are easily adjustable by means of the desired width and amplitude. The width of the gaussian for each behaviour is related to the desired influence radius. To begin, we can use an unitary amplitude in the design phase and let the coordination phase to weight the behaviours. However we have to pay attention to the sign of the gaussian. A positive sign means a repulsive behaviour (high potential), whereas a negative amplitude means an attractive behaviour (low potential).

Thereby, using this simple potential field approach, 6 basic behaviours have been defined:

- *Go to Frontier*: Frontier cells attract the robots since these cells lead the robot to new zones to explore. We can evaluate the potential field as follows:

$$P_1(i,j) = -\sum_{k \in c_f} \exp\left(-\frac{(i-k_i)^2 + (j-k_j)^2}{2\sigma_1^2}\right), \tag{2}$$

  being $P_1(i,j)$ the potential in the cell (i,j) associated to this behaviour, σ_1 the width for this behaviour, c_f the subset of frontier cells, and (k_i, k_j) the coordinates of the cell k.
- *Go to Unexplored Cells*: This behaviour considers the unexplored cells as a point of attraction. This way the less explored areas have a higher priority. The corresponding potential field is:

$$P_2(i,j) = -\sum_{k \in c_u} \exp\left(-\frac{(i-k_i)^2 + (j-k_j)^2}{2\sigma_2^2}\right), \tag{3}$$

  where $P_2(i,j)$ is the potential in the cell (i,j) associated to this behaviour, σ_2 is the width for this behaviour, and c_u is the subset of unexplored cells.
- *Avoid Obstacles*: In order to prevent collisions, the occupied cells repulse the robots. This way, the potential field is evaluated as:

$$P_3(i,j) = \sum_{k \in c_o} \exp\left(-\frac{(i-k_i)^2 + (j-k_j)^2}{2\sigma_3^2}\right), \tag{4}$$

  being $P_3(i,j)$ the potential in the cell (i,j) associated to this behaviour, σ_3 the width for this behaviour, and c_o the subset of occupied cells.
- *Avoid Other Robots*: The purpose of this behaviour is to spread the robots over the environment. This way, the subset c_r of cells, where other robots are situated, repulses the robot using the following potential field:

$$P_4(i,j) = \sum_{k \in c_r} \exp\left(-\frac{(i-k_i)^2 + (j-k_j)^2}{2\sigma_4^2}\right), \tag{5}$$

  where $P_4(i,j)$ is the potential in the cell (i,j) associated to this behaviour and σ_4 is the width for this behaviour. This behaviour is introduced in order to coordinate the actions of the robots. This way each robot explores a different part of the environment.
- *Go to Precise Pose*: In order to achieve a good localization and consequently a good map, the subset of past precise cells c_p attracts the robots. The corresponding potential field is:

$$P_5(i,j) = -\sum_{k \in c_p} \exp\left(-\frac{(i-k_i)^2 + (j-k_j)^2}{2\sigma_5^2}\right), \tag{6}$$

Table 1. Widths chosen for each behaviour

	Behaviour	$\sigma_b (m)$
1	Go to Frontier	4.5
2	Go to Unexplored Cells	1.65
3	Avoid Obstacles	0.15
4	Avoid Other Robots	3.45
5	Go to Precise Pose	3.45
6	Go to Goal	3.45

Table 2. Weights chosen for each behaviour

	Behaviour	λ_b
1	Go to Frontier	3
2	Go to Unexplored Cells	1
3	Avoid Obstacles	100
4	Avoid Other Robots	2
5	Go to Precise Pose	0.5
6	Path Following	5

being $P_5(i, j)$ the potential in the cell (i, j) associated to this behaviour and σ_5 the width for this behaviour.

- *Path Following*: Sometimes, the robots will need to trace a route over a sequence of cells, as for instance when escaping from singularities. We will use this behaviour in order to attract the robot to the next cell in the sequence. Therefore the potential field is:

$$P_6(i, j) = -\sum_{k \in c_g} \exp \left(-\frac{(i - k_i)^2 + (j - k_j)^2}{2\sigma_6^2} \right), \tag{7}$$

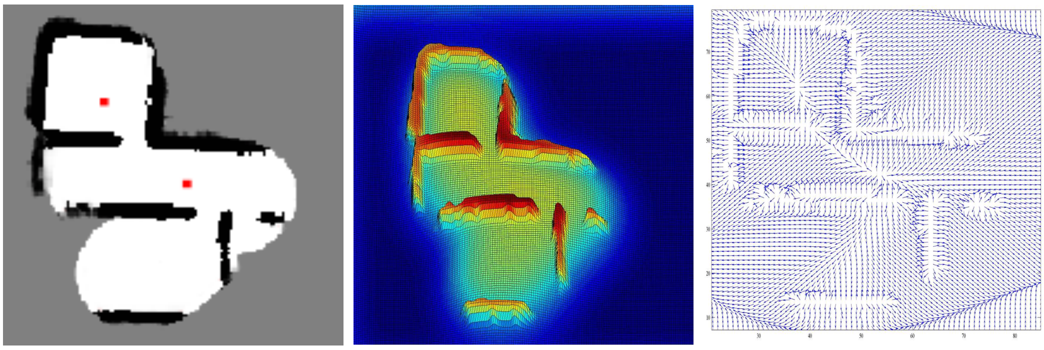
being $P_6(i, j)$ the potential in the cell (i, j) associated to this behaviour, σ_6 the width for this behaviour, and c_g the next goal cell in the path.

As we said, the width of the gaussian for each behaviour is related to the desired influence radius. This radius can be considered as 3σ. This way Table 1 shows the widths selected for each behaviour. These values have been chosen empirically.

3.3. **Behaviour Coordination.** Several techniques can be applied to the fusion of behaviours as for instance subsumption, voting, weighted summation or fuzzy logic [33, 34, 35, 36]. In our case, we have demonstrated experimentally that a simple coordination consisting in the weighted summation of the potential generated by each behaviour is enough when setting the appropriate width (σ_b) for each behaviour. The weight for each behaviour k_i represents its relative importance. Table 2 shows the weights empirically chosen for each behaviour. These constants were obtained after an experimental adjustment process with different scenarios and different initial conditions. This way the global potential field is composed as follows:

$$P = \sum_b e_b \lambda_b P_b, \tag{8}$$

where P_b is the potential field for behaviour b, λ_b is its weight, and e_b is a term that takes only the value 0 or 1 and designates whether the behaviour b is enabled or not.



(a) (b) (c)

Figure 4. Example of potential field. a) Occupancy grid map and other robots positions. b) Corresponding potential field. c) Map of the gradient of the potential field (advance direction)

Figure 4 shows an example of the global potential field and gradient generated taking into account the occupancy grid map and the positions of the robots. It can be observed how the frontiers and the position of the other robots affect the potential field.

However, it is not necessary to evaluate this potential field for each cell of the map. In practice only the local potential field will be computed. We use a small window of 7 × 7 cells centred in the cell where the robot is situated using a $0.15cm$ resolution. Thus, we calculate the potential value of each active behaviour and their weighted addition only for these cells. Furthermore, when the potential for each behaviour in each one of the cells of this local window is calculated, and, in order to guarantee a reactive behaviour working in real time, we can despise the cells that lie out of the radius of influence (3σ) of the behaviour that has the wider gaussian. The desired change of orientation for the robot θ_d will be the one that points to the cell of minimum potential in the proximities of the robot:

$$\theta_d = \arctan\left(\frac{P_y^m - P_y^r}{P_x^m - P_x^r}\right) - \theta_r, \tag{9}$$

where θ_r is the current orientation of the robot, (P_x^r, P_y^r) are the coordinates of the robot in this local potential reference frame, and (P_x^m, P_y^m) are the coordinates of the point of minimum potential. Note that we have a discretised representation of the environment and a discretised representation of the potential field in a local window. However, as the robot is situated inside the central cell of this window, the coordinates (P_x^r, P_y^r) represent its exact continuous position in this reference frame. In the same way, the coordinates of the point of minimum potential (P_x^m, P_y^m) are obtained by means of interpolating the potential values of the contiguous cells. This way we obtain the exact position in a continuous state.

Finally the sequent control laws define the control command given to the robot:

$$\omega = \begin{cases} k_\omega \sigma_d & if \quad k_\omega \sigma_d < \omega_{max} \\ \omega_{max} & else \end{cases} \tag{10}$$

$$\nu = \frac{\nu_{max}}{k_\nu |\sigma_d| + 1} \tag{11}$$

where k_ω, k_ν, ν_{max} and ω_{max} are constants experimentally adjusted for a god control of the differentially driven robot platform.

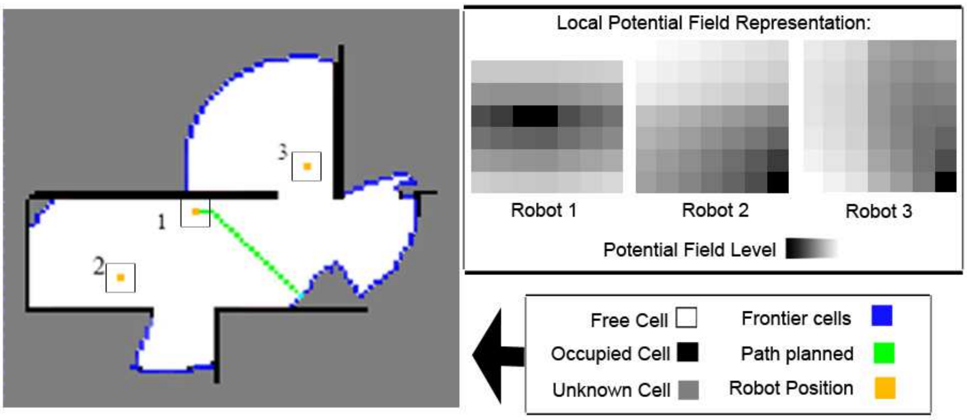


FIGURE 5. Representation of the potential field in the local window during an exploration process with tree robots. Robot 1 detects a local minimum and plans a path to the nearest frontier cell.

3.4. **Behaviour Arbitration.** Not all the behaviours defined in section 3.2 are active all the time. In this sense, we propose an arbitration scheme that decides what behaviours are enabled at each moment. In the case of the integrated exploration problem, we consider 3 possible states that are related to different sets of active behaviours, which are associated to different tasks. Firstly the exploration of the unknown environment, secondly the preservation of a good localization of the robots, and finally, the prevention of getting blocked by a local minimum.

In these situations with several robots working together to explore a common environment, local minima are likely to appear. For instance, the movements of a robot can generate a frontier behind a wall where there is another robot. In this case, the *go to frontier* and *avoid obstacles* behaviours compensate each other generating a local minimum in the resultant potential field at that point. Consequently, this other robot would remain blocked in that position until the frontier that generates the local minimum is removed by other robot. This fact may cause that the robot does not contribute to the exploration for a while or even that the exploration task could not be finished if all the robots get blocked. Figure 5 shows an example of a blocked robot during an exploration process with three robots. The figure shows the shared occupancy grid map with the positions of the three robots. A representation of the local potential field for each robot can be seen on the right of the figure as grey scale images. These images show the evaluation of the discretised potential field in a small window centred in the cell where each robot is situated as explained in Section 3.3. The grey level indicates the potential field. In this way, the darker zones in each image are the zones of lower potential in the surroundings of each robot. We can see in the image for robot 1 that a local minimum is situated in the position of the robot, since the cell of minimum potential appears in the centre of the image. This local minimum is generated by the cancellation of the attractive behaviour to the frontier that is behind the wall with the repulsive behaviour from those cells that are occupied because of that wall. Since the robot travels following the zones of minimum potential, this local minimum blocks the robot.

For that reason, when a local minimum is detected we need a state to let the robot escape from that minimum. This state will consist in following a planned path in order to get the robot out of this minimum. Consequently, we have considered 3 possible states:

TABLE 3. Possible states

	State	Behaviours
A.	Exploration	Go to frontier Go to unexplored areas Avoid other robots Avoid obstacles
B.	Active Localization	Go to precise poses Avoid obstacles
C.	Escape from Local Minimum	Path Following Avoid obstacles

- *A: Exploration*: This state performs a reactive exploration. In this state the behaviours: *Go to Frontier, Go to Unexplored Zones, Avoid Other Robots and Avoid Obstacles* are enabled.
- *B: Active Localization*: This behaviour guides the robot to past precise poses when the robot is poorly localized. It uses the *Go to Precise Pose and Avoid Obstacles.* This is used to reduce the uncertainty in the position of the robots.
- *C: Escape from Local Minimum*: This state is used to escape from a local minimum when it is detected. In this state the robot follows a planned path that leads it to a target cell. The target cell is selected depending on the localization uncertainty. When the uncertainty is low we use the nearest frontier cell, but when the robot is poorly localized the nearest past precise pose cell is selected as target. The active behaviours are *Path Following and Avoid Obstacles.*

Table 3 summarizes the possible states and the behaviours that are enabled in each case. These three states are sequenced in a finite state automata (FSA). Figure 6 shows this FSA for the integrated exploration problem. As we can see, the comparison of the dispersion in the position of the robots with the high (Th_a) and low (Th_b) thresholds triggers between the well localized part of the diagram (upper side) and the poor localized part (lower side). Furthermore, a local minimum detector triggers the *Escape from Local Minimum* state. We need to set a goal for the path followed by the *Escape from Local Minimum* behaviour. However, this goal must be set accordingly with the degree of localization. Consequently, we have separated the *Escape from Local Minimum* state in two cases when the robot is well localized (C) and when the robot is poorly localized (C'). In the first case, the most appropriate action is to plan a path to the nearest frontier cell. In the second case, the path is planned to a past precise pose cell. The completion of the escaping route returns the robot to its previous state.

The detection of local minima is made by analysing the potential field in the neighbourhood of the robot. As can be seen in Figure 5, the cell of minimum potential usually is situated in the border of the local area. However, when the cell of minimum potential appears in the centre of the local window we can identify a local minimum. In this way, when the position of minimum potential in this window is very close to the robot position we consider that the robot is trapped by a local minimum. This detection technique is immediate, as we already have the local potential field. In the case of detecting a local minimum, the robot determines its nearest frontier (or past precise pose) and the shortest path to arrive to it using the Dijkstra's Algorithm [37].

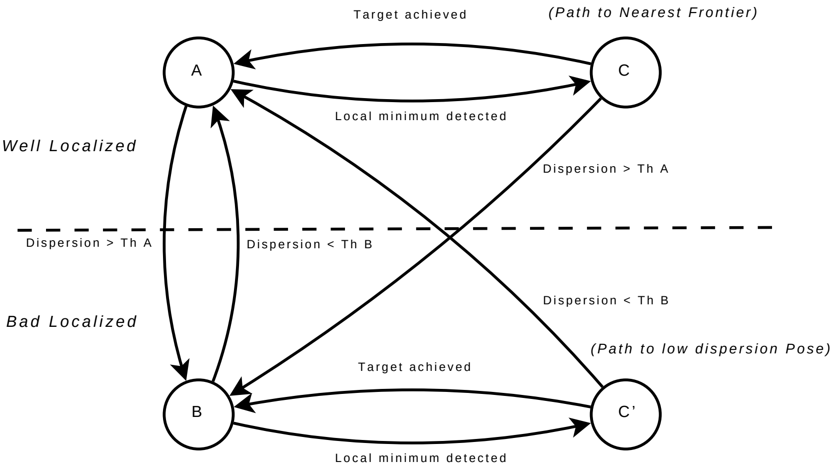


Figure 6. Finite state automata that controls the sequence of active behaviours

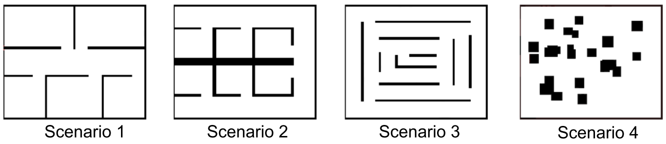


Figure 7. Simulation environments

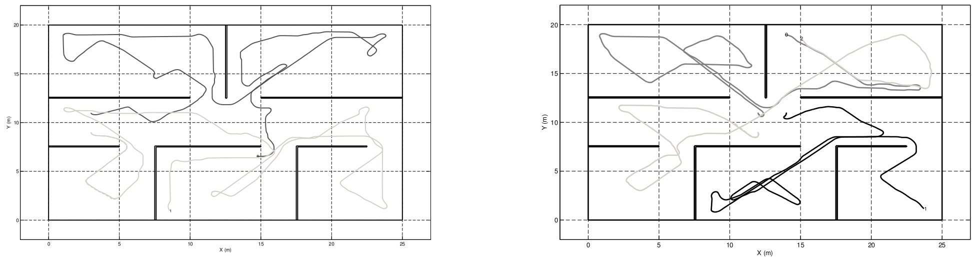
4. EXPERIMENTS.

4.1. **Test bench.** This method has been tested in simulation. Figure 7 shows the scenarios used in the tests. Scenarios that represent hypothetical real places like Scenario 1 or Scenario 2 were chosen. Other artificial scenarios as for instance Scenario 3, which may cause a lot of local minima, or a completely random scene as Scenario 4 were also selected. All the scenarios, with fixed dimensions of $20 \times 25m$, have a predetermined set of approximately 100 landmarks randomly positioned over the obstacles. Each landmark is identified with a visual descriptor.

The simulated robots move with a linear speed limited to $0.05m/s$ and the angular speed limited to $0.03rad/s$. Each robot has a sonar that consists of a set of 8 sensors with a maximum range of $5m$ that cover the front of the robot. These sensors are situated at fixed intervals (with regard to the advance direction) from $\pi/2$ to $-\pi/2rad$. Furthermore the robots have a simulated stereo camera looking forward that is used to detect the landmarks within a range of $8m$. We consider that the visual descriptor can be extracted perfectly. In addition, a perfect data association between the stereo images and the visual landmarks map is obtained.

Regarding the visual SLAM, the number of particles used in the particle filter were proportional to the number of robots N_r. This way we have more particles to represent suitable combinations of the robots paths. Theoretically the number of particles should be exponential with the number of robots. However, using an exponential relationship is computationally not feasible. Thus, considering that the distance travelled will be shorter as the number of robots grows and that possible observations of the same landmarks can improve the results, we have chosen a linear number of particles $M = 500N_r$. The occupancy grid map is obtained with a resolution of $15cm$.

The exploration algorithm presented in this paper was adjusted with the parameters specified in Table 1 and Table 2. The localization thresholds were empirically set to



(a)

(b)

FIGURE 8. Trajectories of 2 robots (a) and 3 robots (b) in scenario 1.

$Th_a = 0.6m$ and $Th_b = 0.3m$. These values are estimated with an experimental tuning process analysing several situations and environments. With these values the robots can return to previously explored areas when they have a high uncertainty in their localization. The level of these values affects the number of times the robot returns to already explored areas. The selection of small thresholds leads to accurate maps as the robot keeps always a lower uncertainty, but increases significantly the exploration time.

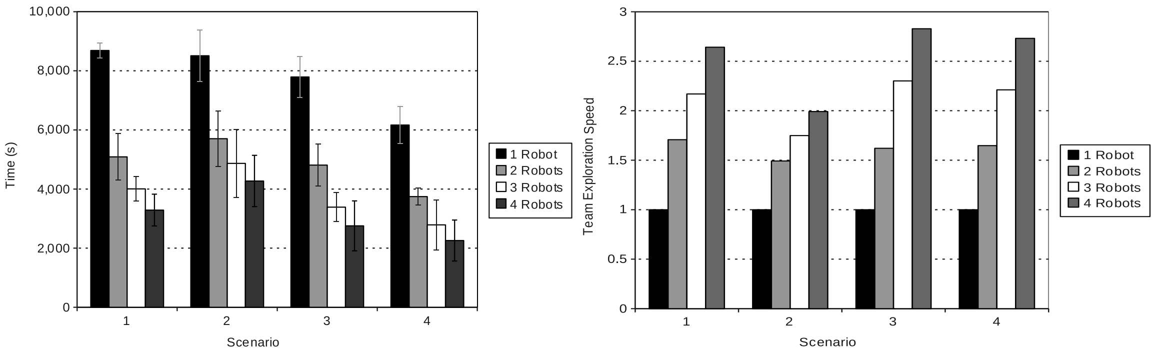
Experiments were performed for each scenario varying the number of robots in the team from a group of 1 to a group of 4 robots. Each experiment was performed 20 times changing the initial positions of the robots. However, the robots always start the exploration in near locations. All the results presented hereafter are the average of all these simulations. To consider that an scenario has been fully explored we need to set an exploration ending condition. In this sense, when there is no remaining frontiers, the robots usually tend to go to a local minimum. In this case, as there are no remaining frontiers, the Dijkstra's Algorithm does not find any frontiers. This way, the case of no frontiers found by the Dijkstra's Algorithm is taken as the exploration ending condition.

The experiments were carried out using a simulation time with a fixed time period of $1s$. That means that independently of the time needed for the calculations we assume that the elapsed time between data acquisition and the new commands given to the robots is fixed. The exploration algorithm to decide the movements of the robots is fast enough to operate in real time at this frequency. However, the map building algorithm is the bottleneck for the whole architecture. When using a large number of robots with a high number of particles and specially when the number of landmarks in the visual map becomes significant the time needed for the calculations is higher than the established period using a $2.6GHz$ processor. However, some improvements could be done as the parallelization of the SLAM algorithm.

Next, we analyse the results in terms of exploration speed, which depends on the quality of robot coordination. Afterwards the quality of the obtained maps will be studied.

4.2. **Multirobot Coordination.** Figure 8 shows an example of the trajectories performed by the team of robots in *Scenario 1*. We can observe that the robots are able to coordinate themselves and they explore different zones separately. The robots try to avoid each other and they spread themselves over the environment. They only work at the same areas at the end of the exploration when few frontiers remain to be explored.

To visualize how good is the coordination we can pay attention to the exploration time. It can be measured as the necessary number of time units needed to complete the exploration. Figure 9(a) shows the average exploration time needed by the robots in each one of the scenarios while changing the number of robots that make up the team. As it can be seen, the exploration time for each scenario reflects their different complexity



(a) (b)

FIGURE 9. Time elapsed to complete the exploration and team exploration speed gain

level. For instance, *Scenarios 1 and 2* need more time, since their structures force the robots to trace back their paths at each dead end. It can also be seen in the same figure how the exploration time changes with the addition of robots. As it was expected, in all the scenarios the exploration time decreases gradually when the number of robots grows.

To effectively measure the coordination independently of the length required for the paths in each scenario, it can be compared the exploration time needed by the team t_n and the exploration time for a single robot t_1. This means that using n robots supposes a gain $g_n = t_1/t_n$. In this sense, a gain $g_n^{PC} = n$ would be expected in a perfect coordination, that is, n robots should do the task n times faster than 1 robot. For a non coordinated algorithm with the robots starting in close positions it is likely that all the robots make very similar paths. That would lead to a gain $g_n^{NC} \approx 1$.

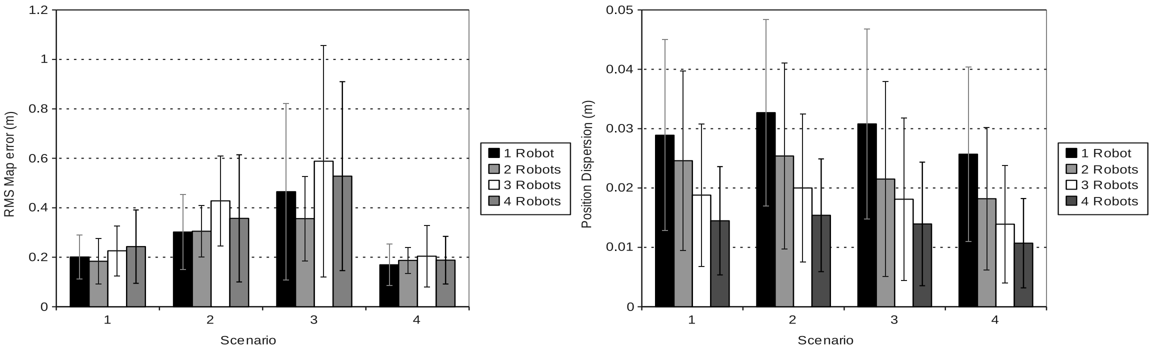
Figure 9(b) shows the exploration gain according to the experiments. We can see that the exploration speed gain is increased as the number of robots grows. It is not a perfect coordination. An important factor to consider here is that the experiments were carried out starting with the robots in a grouped configuration. This fact obviously reduces the possibility to obtain a perfect coordination. Nonetheless, we can see that the different scenarios perform in very different ways. Specially, the second scenario, which needs the robots to travel a long path, has very poor gain with the addition of robots. However, the other scenarios, that have more bifurcations, have a higher gain. This is because in these scenarios the robots can exploit better the robot coordination mechanisms obtaining a significant gain with the addition of robots.

4.3. **SLAM Integration.** The map quality can be evaluated by means of comparing the position of each landmark in the visual map associated to the most probable particle with their real positions in the simulated environment. This error in the map (E^m_{RMS}) can be expressed as the root mean square:

$$E^m_{RMS} = \sqrt{\frac{1}{N}\sum_{l=1}^{N}(\vec{x}^l_m - \vec{x}^l_r)^2}, \tag{12}$$

where N is the number of landmarks detected, $\vec{x}^l_m$ is the 3D position of the landmark l in the estimated map and $\vec{x}^l_r$ is the real 3D position of the landmark l.

Using this metric, the resulting error of the experiments is shown in Figure 10(a). As it can be seen, the error in the visual landmarks map obtained is small in relation with



(a) (b)

Figure 10. Error in the map and dispersion on robot position.

the dimensions of the explored zone. It can be observed that the scenarios that require to travel a long path through unknown zones present a higher error.

Figure 10(b) shows the average dispersion in the robots positions. It can be observed how it is limited with the fixed localization thresholds (Th_a, Th_b). According to the design of the algorithm, the SLAM is considered and integrated into the motion control by means of the active localization state that forces the robot to return to previously explored zones when the uncertainty in the position of the robots is high. This way, the robot maintains a low localization uncertainty, and thus, a low error is committed in the visual map. This is the purpose of the SLAM integration into the exploration algorithm.

As the robots are exploring the same space and sharing their acquired data to build a common map, it could be thought that the error should decrease as the number of robots grows. However, it can be seen in Figure 10(a) that the error does not have a clear dependency with the number of robots. It is worth pointing out that increasing the number of robots affects the algorithm in several ways:

- more observations are added to the system.
- the distance travelled by each robot is shorter.
- each particle is a worse representation of the state of the robots since it has to represent the position of all the robots in the system.

The first two points should improve the quality of the map, however the third point should reduce it. Furthermore, the error depends obviously on the number of particles that was decided to be proportionally set with the number of robots. It can be better understood how the number of particles and the number of robot positions that each particle has to represent affects the system by means of analysing the number of *effective particles*. The *effective particles* of the particle filter are defined as those particles that survive in the resampling step of the filter. The number of *effective particles* should be as great as possible to maintain a good representation of the suitable paths for the robots. In Figure 11 we can see the average number of *effective particles* in the experiments. We can see that, despite the linear increasing of the number of particles with the addition of robots, the number of effective particles does not grow proportionally, and its standard error increases.

Consequently, using small teams of robots, the error is quite stable and it does not increase significantly. Thus, the use of a number of particles proportional to the number of robots is an acceptable approximation with small teams. However, as the number of robots grows, it can be seen a tendency on the number of effective particles to be reduced. In these cases, the capability of the filter to represent suitable positions for all the robots

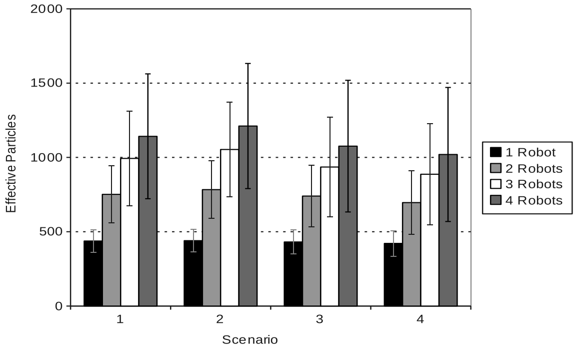


FIGURE 11. Number of effective particles.

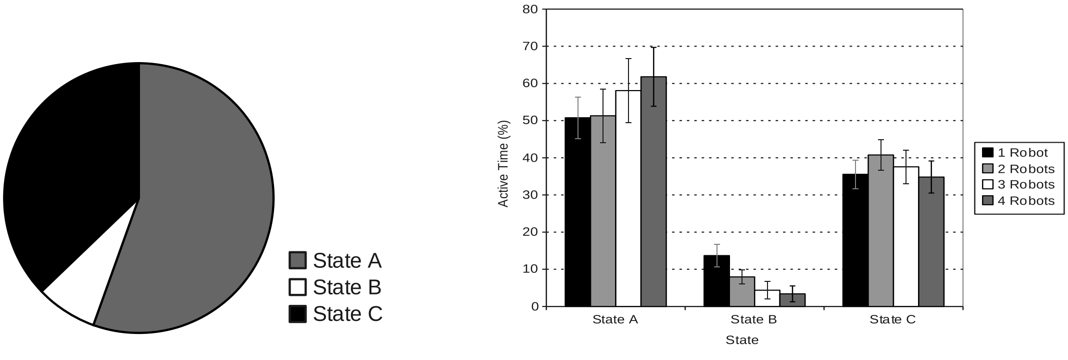
TABLE 4. Classification of exploration techniques

		COORDINATION	
		no coordinated	**coordinated**
SLAM INTEGRATION	**classic**	[8, 15]	[20, 10, 16, 11]
	integrated	[5, 9]	[12] BBMRIE

is low. Therefore, although the exploration algorithm that controls the motion of the robots can be used with a high number of robots, the SLAM algorithm cannot be scaled appropriately in real-time without an exponential increment of the number of particles with the number of robots that is computationally infeasible. Consequently, the use of this SLAM technique is only recommended for small teams.

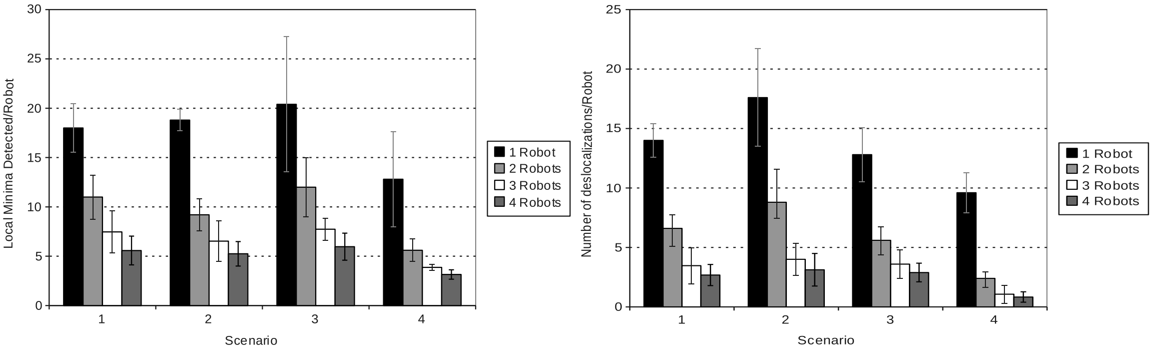
5. **Comparison with other techniques and performance analysis.** Table 4 shows a relative comparison between the Behaviour Based Multi-Robot Integrated Exploration technique (BBMRIE) exposed in this paper and other exploration techniques. In contrast to these other techniques this one can be classified as an integrated and coordinated exploration algorithm. As it was previously explained, the classical exploration approaches [8, 15] always direct the robots trying to maximize the information gain but they do not take into account the quality of their localization. Since the accuracy in the SLAM problem is affected by the trajectories of the robots, it is necessary that the robots integrate somehow the uncertainty in their localization in its motion algorithm. Integrated exploration algorithms [5, 9] deal with this problem. Furthermore, an exploration algorithm should include coordination mechanisms to exploit the work in parallel of a team of mobile robots [20, 10, 16, 11]. However, very few algorithms consider coordination and SLAM integration at the same time [12]. In this paper, it has been presented a new exploration algorithm (BBMRIE) that includes coordination and SLAM integration in a very simple way. As it has been shown in Section 4, it was successfully tested showing a good coordination and a good map quality according to the SLAM integration.

In order to analyse the performance of the algorithm, some statistics of the operation of the FSA are very interesting. Figure 12(a) shows the average percentage of time that the robots are running in each state considering all the scenarios and different team sizes. As we can see, approximately 56% of the time they are working in the *Exploration* state (A). A 7% of the time they are poorly localized and they work in the *Active Localization* state (B). The rest 37% of the time is spent escaping from the local minima detected (C). Figure 12(b) shows how these values change with the number of robots. We can observe that they have a clear dependence with the number of robots. As the number of



(a) (b)

FIGURE 12. FSA activation states



(a) (b)

FIGURE 13. FSA transitions: a) Local minima detected. b) Number of times with poor localization.

robots grows, the dispersion decreases. This was shown in Figure 10(b). Consequently the robots spent less time in *Active Localization* and more time in *Exploration*.

From Figure 13 we can get an idea of the transitions in the FSA. Figure 13(a) shows the number of local minima detected and Figure 13(b) shows the number of times that the high dispersion threshold Th_a is exceeded. In all cases, the number of local minima detected and poor localizations per robot decreases with the number of robots, however the total number of local minima remains approximately constant and the total number of poor localizations is slightly reduced. This can be explained by the lower number of effective particles as the number of robots grows. As the result of these times when the poor localization threshold is exceeded, a 7% of time in average is invested to return to previously explored zones and improve the localization. However, it should be noticed that this is only the time employed in going back, naturally, the robots might come to invest a similar time in going to new frontiers again.

It could be observed from the experiments that the number of local minima detected is considerable and affects significantly the performance of the algorithm. As it has been shown, 37% of the time the robots are following planned paths to escape from local minima. Despite this time is useful because it leads the robots to unexplored frontiers, some time is wasted before when the robot is navigating towards a local minimum until it detects it. This is a point to improve in future works.

6. **CONCLUSIONS AND FUTURE WORKS.** In this paper we have presented a method for multi-robot cooperative integrated exploration. The method is based on the

computation of a set of simple behaviours using potential fields. These behaviours are designed to simultaneously consider both the necessity of rapidly exploring the whole environment and the accuracy of the map building process. In this sense, the method directs the robots to return to previously explored places when the uncertainty on the location becomes significant. This fact improves the quality of the resulting map. The SLAM is carried out with a Rao-Blackwellized particle filter. The known problem of local minima in the potential field methods has been resolved with a strategy of detection and escape. The detection is made analysing the potential field in the surroundings of the robot. The escape from a detected local minimum is made following a planned path to the nearest frontier.

The approach has been tested in simulation. The results show a good coordination between the robots. The exploration time is reduced notably with the number of robots. The statistics show that the robots employ a great part of the time exploring and little time in active localization actions. The error of the maps obtained is relatively small with a small team of robots. In contrast to classical exploration techniques, returning to previously explored zones when the localization uncertainty is high improves the quality of the maps. Despite the motion control algorithm has a good scalability, further work should be done in the scalability of the SLAM for a team of a high number of robots.

As future works, we consider the extension of the approach to real and dynamic environments. Besides, the experimentally design of the parameters can be difficult. In this sense, the addition of techniques to learn automatically the multiple settings of the system will be considered. We have also observed a loss of efficiency caused by the local minima appearance. A hybrid model can be useful considering only a local area free of local minima and using a high level planner. Furthermore, the SLAM technique applied has been proved not to be appropriate for working in real-time with teams of a high number of robots. In that sense, a decentralized SLAM where each robot builds his own map will also be incorporated including techniques for map alignment and fusion. Semi-operated models that integrate the commands expressed by a human operator in the exploration task will also be studied.

Acknolegments. This work has been supported by the Spanish Government (Ministry of Science and Innovation). Projects: 'Sistemas de percepción visual móvil y cooperativo como soporte para la realización de tareas con redes de robots'. Ref.: DPI2007-61197 and 'Exploración integrada de entornos mediante robots cooperativos para la creación de mapas 3D visuales y topológicos que puedan ser usados en navegación con 6 DOF'. Ref.: DPI2010-15308.

REFERENCES

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

[2] I. Rekleitis, R. Sim, G. Dudek, and E. Milios. 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, 2001.

[3] Yuki Tateno Kazutaka Yokota Fairul Azni Jafar, Yasunori Suzuki and Takeshi Matsuoka. Autonomous mobile robot navigation method based on environmental visual features. *ICIC Express Letters,*, 3(3(B)):865–870, 2009.

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

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

[6] Jianguo Zhao Bing Li, Xiaojun Yang and Ping Yan. Minimum time trajectory generation for a novel robotic manipulator. *International Journal of Innovative Computing, Information and Control*, 5(2):369–378, 2009.

[7] B. Yamauchi. Decentralized coordination for multirobot exploration. *Robotics and Autonomous Systems*, 29:111–118, 1999.

[8] A.C. Schultz, W. Adams, and B. Yamauchi. Integrating exploration, localization, navigation and planning with a common representation. *Autonomous Robots*, 6:293–308, 1999.

[9] C. Stachniss, G. Grisetti, and W. Burgard. Information gain-based exploration using rao-blackwellized particle filters. In *In Proceedings of Robotics: Science and Systems (RSS'05)*, Cambridge, MA, USA, 2005.

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

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

[12] B. Tovar, L. Munoz-Gomez, R. Murrieta-Cid, M. Alencastre-Miranda, R. Monroy, and S. Hutchinson. Planning exploration strategies for simultaneous localization and mapping. *Robotics and Autonomous Systems*, 54(4):314–331, 2006.

[13] B. Yamauchi. 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, 1997.

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

[15] H. H. Gonzalez-Banos and J. C. Latombe. Navigation strategies for exploring indoor environments. *International Journal of Robotics Research*, 21(10), 2002.

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

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

[18] R. Arkin. *Behavior Based Robotics*. MIT Press, 1998.

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

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

[21] M. Julia, A. Gil, L. Paya, and O. Reinoso. Local minima detection in potential field based cooperative multi-robot exploration. *International Journal of Factory Automation, Robotics and Soft Computing*, 3, July 2008.

[22] Y. Xiaoping and T. Ko-Cheng. 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, California, 1997.

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

[24] R. Sim, G. Dudek, and N. Roy. On line control policy optimization for minimizing map uncertainty during exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'04)*, New Orleans, LA, USA, 2004.

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

[26] Martijn N. Rooker and Andreas Birk. Multi-robot exploration under the constraints of wireless networking. *Control Engineering Practice*, 15(4):435–445, 2007.

[27] Genci Capi. Application of recurrent neural controllers for robot complex task performance. *International Journal of Innovative Computing, Information and Control*, 5(5):1171–1178, 2009.

[28] A. Gil, O. Reinoso, M. Ballesta, and M. Julia. Multi-robot visual slam using a rao-blackwellized particle filter. *Robotics and Autonomous Systems*, 58(1):68–80, 2010.

[29] M. Ballesta, A. Gil, O. Reinoso, and O. Martinez-Mozos. Evaluation of interest point detectors for visual slam. *International Journal of Factory Automation, Robotics and Soft Computing*, 2008-4:86–95, 2008.

[30] M. Montemerlo and S. Thrun. 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, 2003.

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

[32] Xinde Li, Xinhan Huang, Jean Dezert, Li Duan, and Min Wang. A successful application of dsmt in sonar grid map building and comparison with dst-based approach. *International Journal of Innovative Computing, Information and Control*, 3(3):539–549, 2007.

[33] R. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14– 23, 1986.

[34] J. Rosenblatt. Damn: A distributed architecture for mobile navigation. *Journal of Experimental and Theoretical Artificial Intelligence*, 9(2-3):339–360, 1997.

[35] P. Althaus and H. I. Christensen. Behaviour coordination in structured environments. *RSJ Advanced Robotics*, 17(7):657–674, 2003.

[36] A. Safiotti. Fuzzy logic in autonomous robotics: behavior coordination. In *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems*, Barcelona, Spain, 1997.

[37] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.