Coordinated multi-agent exploration

Abraham Sánchez L. and Alfredo Toriz P.

Facultad de Ciencias de la Computación, BUAP
14 Sur esq. San Claudio, CP 72570 Puebla, Pue., México
`asanchez@cs.buap.mx, alfredot@hotmail.com`

Abstract. Many successful robotic systems use maps of the environment to perform their tasks. In this paper, we propose a cooperative exploration strategy for multi-agent robots. This proposal is a parallelization of the basic SRT method, the following functionalities were added to it: cooperation to increase the efficiency, coordination to avoid conflicts and communication to cooperate and to coordinate. The goal in robot exploration must be to minimize the overall exploration time, and multiple robots produce more accurate maps by merging overlapping information that helps to stabilize the sensor uncertainty and to reach the goal. We present simulation results to show the performance of the proposed technique.

1 Introduction

Although most mobile robotic systems use a single robot that only operates in its environment, a number of researchers have considered the advantages and disadvantages of the potential use of a group of robots that cooperate for the accomplishment of a required task [1], [2], [3]. Multi-agent systems (MAS), may be regarded as a group of entities called agents, interacting with one another to achieve their individual as well as collective goals.

The research domain of multi-agent robot systems can be divided into subdomains according to the task given to the robot group [4]. At present well-studied subdomains are motion planning, formation forming, region-sweeping and combinations of the foregoing. We focused this paper in the region-sweeping task. In the region-sweeping task, one can consider two activities.

In the first activity, a group of robots receives the order to explore/map an unknown region. The goal is to obtain a detailed topography of the desired area. In most exploration approaches, the boundary between know and unknown territory (the frontier) is used in order to maximize the information gain. In [5], the robots merge the acquired information in a global grid-map of the environment, from which the frontier is extracted and used to plan the individual robot motions. The frontier-based approach lacks of an arbitration mechanism preventing the robots from approaching the same frontier region. The approach presented in [3] proposed to negotiate robot targets by optimizing a utility function which takes into account the information gain of a particular region, the cost of reaching it and the number of robots currently heading there. The utility of a particular frontier region from a viewpoint of relative robot localization and the accuracy of map merging were considered in [6]. The incremental deployment algorithm considers that the robots approach the frontier while they retain visual contact with each other [7]. A multi-robot architecture proposed in [8] guide the exploration by a market economy, whereas [9] proposes a centralized approach which uses a frontier-based search and a bidding protocol assign frontier targets to the robots.

Closely related to the exploring/mapping activity, the second one is called complete coverage, where the robots have to move over all of the free surface in the space.

Since exploration algorithms are already devised for a single robot it seems straightforward to divide the area to be explored into disjunct regions, each of which is assigned to a single robot. The robots communicate to each other the area they have explored so that no part of the free space will be explored twice unnecessarily. At no point during the task are the robots trying to form a fixed formation. Each robot explores a different part of the unknown region and sends its finding to a central device which combines the data received from the robots into one global map of the area.

This paper presents a strategy to explore an unknown environment by multi-agent robots. The strategy is a parallelization of the SRT (Sensor-based Random Tree) method, which was presented in [10]. The SRT method, is an exploration method based on the random generation of robot configurations within the local safe area detected by the sensors. A data structure is created, which represents a roadmap of the explored area with an associated safe region (SR). Each node of the SRT consists of a free configuration with the associated local safe region (LSR) as reconstructed by the perception system; the SR is the union of all the LSRs. The LSR is an estimate of the free space surrounding the robot at a given configuration. In general, its shape will depend on the sensor characteristics but may also reflect different attitudes towards perception (see for example [11] for an interesting extension of the SRT method). The extension of the SRT method to multi-agent robots is essentially a parallelization of the basic method, we called this extension, the Multi-SRT method. A decentralized cooperation mechanism and two coordination mechanisms are introduced to improve the exploration efficiency and to avoid conflicts. The basic steps of the exploration approach are presented in Section II. Simulation results in different environments are discussed in Section III. Finally, conclusion and future work are detailed in Section IV.

2 Cooperative exploration

MAS may be comprised of homogeneous or heterogeneous agents, it is considered as crucial technology for the effective exploitation of the increasing availability of diverse of heterogeneous and distributed on-line information sources. MAS is a framework for building large, complex and robust distributed information processing systems which exploit the efficiencies of organized behavior. Team-work and communication are two important processes within multi-agent robots designed to act in a coherent and coordinated manner.

Consider a population of n identical robots. Each robot is equipped with a ring of range finder sensor or a laser range finder, the sensory system provides the local safe region (LSR) $S(q)$. The robots move in a planar workspace, i.e., $\mathbb{R}^2$ or a connected subset of it; the assumption of planar workspace is not restrictive, 3D worlds are admissible as long as the sensory system allow the reconstruction of a planar LSR for planning the robot motion. Each robot is a polygon or another shape subject to non-holomic constraints. The robot also knows its configuration q, one can eliminate this assumption by incorporating a localization module in the method. The robots know its ID number and each robot can broadcast within a communication range R_c the information stored in its memory (or relevant portions of it) at any time. The robot ID number is included in the heading of any transmission. The robot is always open for receiving communication from other robots inside R_c.

The design of the cooperative exploration strategy proceeds from the parallelization of the basic SRT method, each robot builds one or more partial maps of the environment, organized in a collection of SRTs. Each node of an SRT represents a configuration q which was visited by at least one robot, together with the associated Local Safe Region $S(q)$. An arc between two nodes represents a collision-free path. The tree is incrementally built by extending the structure in the most promising direction via a biased random mechanism. The presence of other robots in the vicinity is taken into account at this stage in order to maximize the information gain and guarantee collision avoidance.

The exploration algorithm for each robot is shown in Figure 1. First, the procedure BUILD_SRT is executed, i.e., each robot builds its own SRT, $\mathcal{T}$ is rooted at its starting configuration q_{init}. This procedure terminates when the robot can not further expand $\mathcal{T}$. Later, the robot executes the SUPPORT_OTHERS procedure, this action contributes to the expansion of the SRTs that have been built by others robots. When this procedure finishes, the robot returns to the root of its own tree and finishes its exploration.

```
BUILD Multi-SRT($q_{init}$)
    $\mathcal{T}.init(q_{init})$
    BUILD_SRT($q_{init}.\mathcal{T}$);
    SUPPORT_OTHERS($q_{init}$);
```

Fig. 1. The Multi-SRT algorithm.

The procedure BUILD_SRT is shown in Figure 2. In each iteration of the BUILD_SRT, the robot uses all available information (partially collected by itself and partially gained through the communication with other robots) to identify the group of engaged robots (GER), i.e. the other robots in the team with which cooperation and coordination are adequate. This is achieved by the construction of the first group of pre-engaged robots (GPR), or robots that are candidates to be members of the GER, and are synchronized with them (BUILD_AND_WAIT_GPR). Then, the robot collects data through its sensory systems, it builds the current LSR (PERCEIVE) and updates its own tree $\mathcal{T}$. The current GER can now be built (BUILD_GER). At this point the robot processes its local frontier (the portion of its current LSR limit leads to areas that are still unexplored) on the basis of $\mathcal{T}$ as well as any other tree $\mathcal{T}_i$ gained through communication and stored in its memory (LOCAL_FRONTIER).

If the local frontier is not empty, the robot generates a random configuration contained in the current LSR and headed towards the local frontier, if not, the target configuration is fixed to the node father with a backward movement (PLANNER). If the GER is composed only by the same robot, the robot moves directly to its target. Otherwise, the paths advanced by the robot in the GER are checked for mutual collisions, and classified in feasible and unfeasible paths (CHECK_FEASIBILITY). If the subset $\mathcal{G}_u$ of robots with unfeasible paths is vacuum, a coordination stage takes place, perhaps, confirming or modifying the current target of the robot (COORDINATE). In particular, the motion of the robot can be banned by simply readjusting the target to the current configuration. Then, the function MOVE_TO transfers the robot to the target (when this is different from q_{act}). The loop is repeated until the condition in the output line 15 is verified: the robot is unable to expand the tree $\mathcal{T}$ (no local frontiers remaining) and therefore it has to move back to the root of its SRT.

```
BUILD_SRT($q_{init}, \mathcal{T}$)
1  $q_{act} = q_{init}$;
2  do
3    BUILD_AND_WAIT_GPR();
4    $S(q_{act}) \leftarrow$ PERCEIVE($q_{act}$);
5    ADD($\mathcal{T}, (q_{act}, S(q_{act}))$);
6    $\mathcal{G} \leftarrow$BUILD_GER();
7    $\mathcal{F}(q_{act}) \leftarrow$ LOCAL_FRONTIER($q_{act}, S(q_{act}), \mathcal{T}, \bigcup \mathcal{T}_i$);
8    $q_{target} \leftarrow$ PLANNER($q_{act}, \mathcal{F}(q_{act}), q_{init}$);
9    if $q_{target} \neq NULL$
10     if $|\mathcal{G}| > 1$
11       $(\mathcal{G}_f, \mathcal{G}_u) \leftarrow$ CHECK_FEASIBILITY($\mathcal{G}$);
12       if $\mathcal{G}_u \neq \emptyset$
13         $q_{target} \leftarrow$ COORDINATE($\mathcal{G}_f, \mathcal{G}_u$);
14     $q_{act} \leftarrow$ MOVE_TO($q_{target}$);
15 while $q_{target} \neq NULL$
```

Fig. 2. The BUILD_SRT procedure.

We detail the most important stages of the BUILD_SRT procedure [12]:

Construction of GPR/GER. At the start of BUILD_SRT, the robots are stationary and need to identify the other robots whose LSRs may overlap with its own, in order to cooperate (optimize the exploration) and coordinate (avoid conflicts) with them. The other robots may be stationary as well (in this case, their targets coincide with the current configuration) or moving to the target, therefore a synchronization phase is necessary.

Two robots are GPR-coupled if the distance between their target configurations is less than $2R_p$, i.e. twice the range of perception of the sensorial system. The GPR of the robot is then built by grouping together all robots and connecting a chain of coupled GPRs (see Figure 3, left). To achieve synchronization, the GPR is calculated and updated until all its members are stationary (BUILD_AND_WAIT_GPR). The communication range, R_c, clearly plays an important role in the construction of the GPR; for this method two cases have been considered, according to the communication range:

1. **Limited communication range**. Since the maximum distance between the robot and any other robot GPR-coupled is $3R_p$ (the other robot can still be moving to its target, which however could not be further away than R_p of the current configuration) , it is sufficient to assume that $R_c \geq 3R_p$ to ensure that the GPR accounts for all the robots that are candidates to belong to the GER.
2. **Unlimited communication range**. Given the nature of this communication, the robot always knows the status of the other robots, and therefore will know which robots are candidates to be GPR-coupled; the distance to be GPR-coupled as in the above case will remain $3R_p$. In this way, as in the previous case, it is ensured that the GPR accounts for all the robots that are candidates to belong to the GER.

The synchronization phase ensures that all robots in the GPR are stationary when the GER is processed. The GER is a symmetric structure, this is the same for all robots in the group. The GER is a cornerstone in our approach, as it identifies a group of robots that, in view of its vicinity, spontaneously agree to cooperate and coordinate with each other on a temporary basis.

Frontier extraction. Once the robot has been synchronized with its GER, the procedure LOCAL_FRONTIER is called to process the portion of the limits of the LSR $S(q_{act})$, leading to areas that are unexplored according to the information available. To this end, the robot uses its own tree $\mathcal{T}$ as well as any other tree stored in its memory and received by present or past communications. To find promising directions in $S(q_{act})$, its boundary is divided into arcs with obstacles, free arcs and frontier arcs (see Figure 4).

Planner. The planner determines the new target configuration on the basis of the local an open frontiers associated to the current node. The open frontier of a tree (subtree) is the sum of the lengths of the local frontiers associated to its nodes. If the local frontier $\mathcal{F}(q_{act})$ of the current LSR is not empty, the planner generates a random configuration in the direction of $\mathcal{F}(q_{act})$. Thanks to synchronization done by the function BUILD_ AND_WAIT_GPR, all the robots in a GER plan at the same time, and therefore the cooperation mechanism intrinsic to the definition of the local frontier is strengthened throughout the GER. This agreement attempt is performed without any centralized decision module.

The possibility that the robot is subject to non-holonomic constraints is considered by the function MOVE_TO, which is responsible for generating feasible paths. The controllability of the robot ensures that any goal configuration in the LSR can be reached by paths that are feasible in the LSR.

GER path feasibility check. Although the robot's local frontier can not belong to the LSR of another robot of the GER, the two prospective paths can still intersect. The function CHECK_FEASIBILITY verifies whether the prospective paths of the robots in the GER $\mathcal{G}$ are all simultaneously feasible or not. To this end, all pairs of paths that intersect with others are identified, and the corresponding robots stored in the unfeasible subset $\mathcal{G}_u$ of the GER. The remaining robots constitute the feasible subset $\mathcal{G}_f$ of the GER.

Coordination. If the subset $\mathcal{G}_u$ of robots with unfeasible paths is not vacuum, the coordination function is invoked. The first step is to choose a master robot within $\mathcal{G}$. This can be complemented in many ways through a deterministic procedure known by all the robots, for example, the robot with the highest ID number can be elected. Two cases are possible then:

1) If the robot is the master, it invokes an ORGANIZE function, whose task is to rearrange the vector $\mathcal{Q}_g$ that contains the targets of the robots in the GER and obtain a feasible collective motion. Here, the change may mean whatever, simply accepting or readjusting the target of a robot to the current configuration (i.e., authorizing/prohibiting the motion) or adding a third option, for example, changing to a new target.

2) If the robot is not the master, it enters in a waiting phase, which ends with the receipt of a specified signal from the master.

The final operation is to retrieve and return the robot's (possibly modified) own target from $\mathcal{Q}_g$.

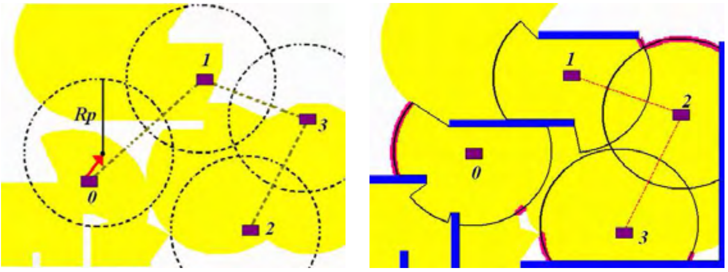

Fig. 3. An example of GPR/GER. At the left, the GPR of robot 1 consists of robots 0, 1, 2 and 3; robot 0 is still moving towards its target point, while robots 1, 2, 3 are stationary. At the right, once the LSR have been computed, only robots 1, 2 and 3 belong to the GER of robot 1 since their LSRs overlap in pairs.

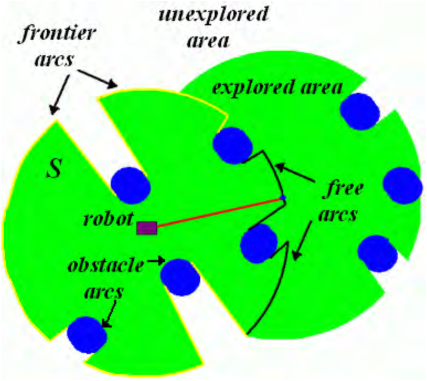

Fig. 4. Definition of frontier, free and obstacle arcs in the Local Safe Region S.

The procedure SUPPORT_OTHERS (see Figure 5) can be divided into two major phases, which are repeated over and over again. In the first phase, the robot picks another robot to support it in his exploration, or, more precisely, another tree that helps it to expand (there may be more than one robot acting on a single tree). In the second phase, the selected tree is reached and the robot tries to expand it, tying subtrees constructed by the procedure BUILD_SRT. The main cycle is repeated until the robot has received confirmation that all the other robots have completed their exploration.

The loop is repeated until the robot has received confirmation that all the other robots have finished their exploration. At the beginning, the robot collects in a set $\mathcal{I}$ the trees belonging to $\bigcup \mathcal{I}_i$ that may require support for expansion. In particular, defining the open frontier of a tree (subtree) as the sum of the lengths of local frontiers associated with its nodes, a tree $\mathcal{T}_i$ is put in $\mathcal{I}$, if its open frontier is at least equal to a constant $\bar{F}$ multiplied by the number of robots that are active in $\mathcal{T}_i$, according to the most recent information available (ACTIVE_ROBOTS). If $\mathcal{I}$ is not empty, the robot selects a particular tree $\mathcal{T}_s$ of $\mathcal{I}$, according to some criterion (for example, the tree with the closest root, or the most recent update), and move to its root. Once there, the robot begins a subtree expansion using the BUILD_SRT procedure. During this process, the robot keeps on trying to add subtrees to $\mathcal{T}_s$ until it has returned to the root of $\mathcal{T}_s$ and its open frontier is zero. At this point, the robot returns to the root of its own tree (i.e. its start configuration) and becomes available to support the expansion of other trees. This phase is only used when an unlimited communication range is handled, because the robot who is providing assistance can count with updated information of the other robots and its last states at any time, in contrast with a limited communication range with probably partial and not updated information.

```
SUPPORT_OTHERS($q_{act}$)
do
  for $i = 1$ to $n$
    if OPEN_FRONTIER($\mathcal{T}_i$) $\geq \bar{F}\cdot$ ACTIVE_ROBOTS($\mathcal{T}_i$)
      ADD($\mathcal{T}, \mathcal{T}_i$);
  $\mathcal{T}_s \leftarrow$ SELECT($\mathcal{T}$);
  if $\mathcal{T}_s \neq$ NULL
    $q_{act} \leftarrow$ TRANSFER_TO($\mathcal{T}_s.root$);
    BUILD_SRT($q_{act}, \mathcal{T}_s$);
    $q_{act} \leftarrow$ TRANSFER_TO($\mathcal{T}.root$);
while EXPLORATION_RUNNING()
```

Fig. 5. The SUPPORT_OTHERS procedure.

3 Simulation results

In order to illustrate the behavior of the Multi-SRT exploration approach, we present two strategies, the Multi-SRT_Radial and the Multi-SRT_Star. The strategies were implemented in Visual C++ V. 6.0, taking advantage of the MSL library's[1] structure and its graphical interface that facilitates to select the algorithms, to visualize the working environment and to animate the obtained path. The library GPC developed by Alan Murta was used to simulate the sensors perception systems[2].

The tests were performed on an Intel © Pentium D processor-based PC running at 2.80 GHz with 1 GB RAM. One can consider two possible initial deployments of the robots. In the first, the robots are initially scattered in the environment; and in the second, the exploration is started with the robots grouped in a cluster. Since the Multi-SRT approach is randomized, the results were averaged over 20 simulation runs. Figure 6 illustrates the environments used for the simulation part. The first is a square region with a garden-like layout, where each area can be reached from different access points. The second is also a square, it contains many obstacles of different shapes.

Exploration time for teams of different cardinality are shown in Figure 7, both in the case of limited and unlimited communication range. In theory, when the number of robots increases, the exploration time would quickly have to decrease. This affirmation is fulfilled in the case of the scattered start; note however that, in the case of the clustered start, there are examples where this affirmation is not verified. We consider that an increment of the number of evenly deployed robots corresponds to a decrement of the individual areas they must cover. In the case of a limited communication range, when the robots are far apart at the start, they can exchange very little information during the exploration process.

---

[1] http://msl.cs.uiuc.edu/msl/

[2] http://www.cs.man.ac.uk/~toby/alan/software/

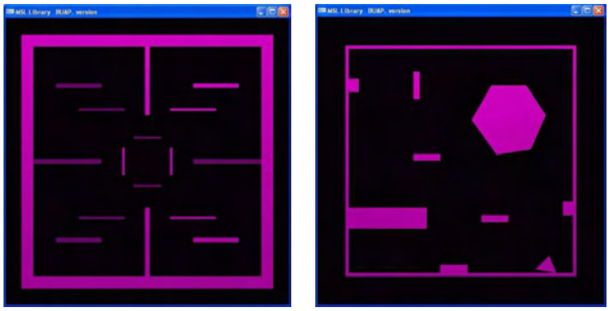

Fig. 6. Environments used for the tests of the Multi-SRT.

The total travelled distance increases with the number of robots because more robots try to support the others in their expansion.

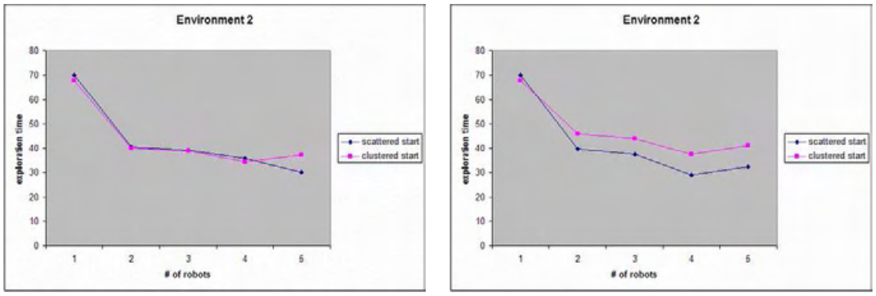

Fig. 7. Environment 2 exploration with scattered and clustered start. To the left with unlimited communication range and in the right with limited communication range.

Figures 9 and 10 show the Multi-SRT and the explored region for the environment 1 with a team of 5 and 10 robots in the case of unlimited communication range, also Figure 8 illustrates the SRT and the explored region for the environment 1 with 1 robot. We can see the difference when the robots are evenly distributed at the start or are clustered. At the end, the environment has been completely explored and the SRTs have been built. In these figures, we can observe that each robot built its own SRT and when one of them finished, this entered the support phase.

If we compare the exploration times in the three last figures, we can affirm that to greater number of robots, the exploration time decreases. If we also observe the placed figure in the center of both figures 9 and 10, one can note that the trees are united, this indicates that the support phase took place. Single robots can not produce accurate maps like multi-robots. The only advantage of using a single robot is the minimization of the repeated coverage.

Fig. 8. The SRT and explored regions with one single robot. Time = 101.408 secs with 112 nodes.

Fig. 9. Environment 1 with 5 robots. The Multi-SRT and explored regions with scattered start. Time = 54.348 secs with 141 nodes.

Fig. 10. Environment 1 with 10 robots. The Multi-SRT and explored regions with scattered start. Time = 26.658 secs with 32 nodes.

However, even though repeated coverage among the robots decreases the mission's efficiency, some amount of repeated coverage is a desirable situation for better efficiency. Additionally, this better efficiency can be achieved by coordination among robots.

The two strategies (Multi-SRT Star and Multi-SRT Radial) were compared through simulations. We used the same environment to prove the efficiency of Multi-SRT Radial over Multi-SRT Star and the same free parameters.

The authors in [10] presented a method called SRT-Star, which involves a perception strategy that completely takes the information reported by the sensor system in all directions; S is a region with the star form because of the union of several 'cones' with different radii each one. Espinoza et al., presented in [11] an interesting extension of the SRT method. The SRT-Radial strategy proposed in this work takes advantage of the information reported by the sensors in all directions, to generate and validate configurations candidate through reduced spaces without the identification of the cone.

Figure 11 above presents the explored regions and safe region obtained with the Multi-SRT Star strategy for the environment 2, for a team of 7 robots in the case of unlimited communication range. On the other hand, Figure 11 down illustrates the explored regions and the safe region with Multi-SRT Radial. The final number of nodes in the tree and the exploration time are much smaller with Multi-SRT Radial that with Multi-SRT Star.

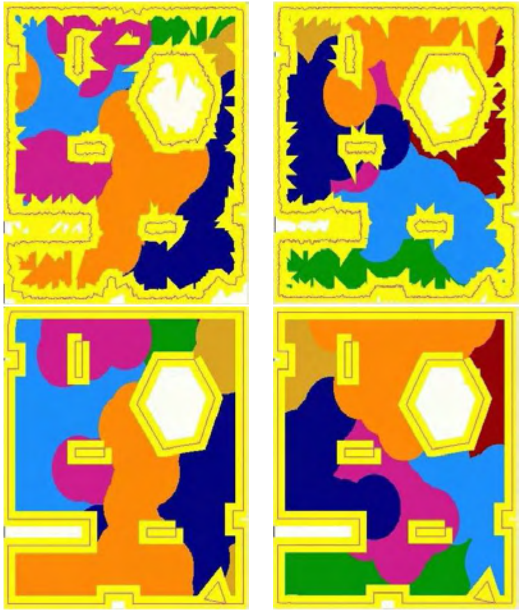

Fig. 11. Environment 2. A comparative between Multi-SRT Star and Multi-SRT Radial.

Architectures for multi-robot exploration are usually classified as centralized and decentralized or distributed. In centralized architectures a central entity plans the actions of all the robot team, while in the decentralized approaches each robot makes use of local information to plan its exploration. Our approach is a decentralized cooperative exploration strategy for mobile robots, its coordination mechanisms are used to guarantee exploration efficiency and avoid conflicts.

Decentralization causes serious problems, such as conflicts among the agents (robots) and their respective goals. This is because the knowledge contained in each agent might be incomplete, and goals of agents might be in conflict. Therefore, conflict resolution is a critical and implicit problem in MAS.

The local coordination procedure implemented in our work guarantees that the collective motion of the robots is feasible from the collision viewpoint. The approach does not need a central supervision. The selection of exploration actions by each robot is spontaneous and it is possible on the basis of the available information.

4 Conclusions and future work

Multi-robot systems are emerging as a the new frontier in Robotics research, posing new challenges and offering new solutions to old problems. Multi-robot systems are not a collection of robots performing a once-for-ever fixed task in a settled and static environment. They are collections of interacting, cooperating autonomous agents with physical embodiment that impose restrictions on what they can do, but also give them power to do some specific things. Thus the paradigms of Multi-Agent and Multi-robot systems are somehow related and the recognition of this parallelism may foster new avenues for research and solutions.

We have presented an interesting approach for cooperative exploration based on the SRT-Radial. The Multi-SRT considers two decentralized mechanisms of cooperation at different levels. The first simply consists in making an appropriate definition of the local frontier that allows each robot to plan its motion towards the areas apparently unexplored for the rest of the team. The second allows a robot that has finished with its individual exploration phase, to support others robots in their exploration task. Additionally, we compared Multi-SRT Radial strategy with Multi-SRT Star strategy, the results obtained with our proposal are more interesting.

Exploration and localization are two of the capabilities necessary for mobile robots to navigate robustly in unknown environments. A robot needs to explore in order to learn the structure of the world, and a robot needs to know its own location in order to make use of its acquired spatial information. However, a problem arises with the integration of exploration and localization. A robot needs to know its own location in order to add new information to its map, but a robot may also need a map to determine its own location. Most exiting localization approaches refer to the single robot case. This means a posture of one robot is decided by its own sensor data, the robot should have an expensive sensor that can measure the robot pose in the global frame. If a robot can use the sensor data of other robots, a cheap sensor system can be used. This reveals the importance of cooperative localization which estimates the pose of each robot by fusing information obtained from the other robots.

The integration of a localization module into the exploration process based on SLAM techniques will be an interesting topic for a future research. We can also consider an extension of the Multi-SRT exploration method, where the robots constantly maintain a distributed network structure.

References

1. Y. Cao, A. Fukunaga and A. Kahng, “Cooperative mobile robotics: Antecedents and directions”, *Autonomous Robots*, Vol. 4, (1-23) 1997
2. G. Dudek, M. Jenkin, E. Milios and D. Wilkes, “A taxonomy for multi-agent robotics”, *Autonomous Robots*, Vol. 3, (375-397) 1996
3. W. Burgard, M. Moors, and F. Schneider, “Collaborative exploration of unknown environments with teams of mobile robots”, *Plan-Based Control of Robotic Agents, LNCS*, Vol. 2466, (2002)
4. Ota J, “Multi-agent robot systems as distributed autonomous systems”, *Advanced Engineering Informatics*, Vol. 20, (2006) 59-70
5. B. Yamauchi, “ Decentralized coordination for multirobot exploration”, *Robotics and Autonomous Systems*, Vol. 29, (1999) 111-118
6. J. Ko, B. Stewart, D. Fox, K. Konolige and B. Limketkai, “A practical, decision-theoretic approach to multi-robot mapping and exploration”, *IEEE Int. Conf. on Intelligent Robots and systems*, (2003) 3232-3238
7. A. Howard, M. Mataric and S. Sukthane, “An incremental deployment algorithm for mobile robot teams”, *IEEE Int. Conf. on Intelligent Robots and Systems*, (2002) 2849-2854
8. R. Zlot, A. Stenz, M. Dias and S. Thayer, “Multi-robot exploration controlled by a market economy”, *IEEE Int. Conf. on Robotics and Automation*, (2002) 3016-3023
9. R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun and H. Younes, “Coordination for multi-robot exploration and mapping”, *17th Conf. of the American Association for Artificial Intelligence*, (2000) 852-858
10. G. Oriolo, M. Vendittelli, L. Freda and G. Troso, “The SRT method: Randomized strategies for exploration”, *IEEE Int. Conf. on Robotics and Automation*, (2004) 4688-4694
11. J. Espinoza L., A. Sánchez L. and M. Osorio L., “Exploring unknown environments with mobile robots using SRT_Radial”, *IEEE Int. Conf. on Intelligent Robots and Systems*, (2007) 2089-2094
12. A. Toriz Palacios, Estrategias probabilisticas para la exploración coopearativa de robots móviles, *Master Thesis*, FCC-BUAP (in spanish), 2007