Coordinated Multi-agent Exploration

Alfredo Toriz P., Abraham Sánchez L.[+] and Maria A. Osorio L.[•]

LIRMM - UMR55606 CNRS Montpellier, France
[+] Facultad de Ciencias de la Computación, BUAP
[•]Escuela de Ingeniería Química, BUAP
alfredot@hotmail.com, asanchez, aosorio@cs.buap.mx

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 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 utility of a particular frontier region from a viewpoint of relative robot localization and the accuracy of map merging were considered in

© G. Sidorov, M. Aldape, M. Martinez, S. Torres. (Eds.)
Advances in Computer Science and Engineering.
Research in Computing Science 42, 2009, pp. 189-200

Received 25/02/09
Accepted 27/04/09
Final version 04/05/09

[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 [10], [11]. Generating maps is one of the fundamental tasks of mobile robots. Many successful robotic systems use maps of the environment to perform their tasks. The questions of how to represent environments and how to acquire models using this representation therefore is an active research area, see [12] for an excellent overview.

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 [13]. The extension of the SRT method to multi-agent robots is called, 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. Teamwork and communication are two important processes within multi-agent robots designed to act in a coherent and coordinated manner [2], [4].

An extensive amount of research has been carried out in the areas of localization, mapping and exploration for single autonomous robots, but only fairly recently has this been applied to multi-robot teams [12]. In addition, nearly all of this research has taken an existing algorithm developed for single-robot mapping, localization, or exploration, and extended it to multiple robots, as opposed to developing a new algorithm that is fundamentally distributed [1], [5], [14]. An interesting exception is some of the work in multi-robot localization, which takes advantage of multiple robots to improve positioning accuracy beyond what is possible with single robots [15]. As in the case with single-robot approaches, the research into the multi-robot version can be described using familiar categories and using either range sensors (such as sonar or laser) or vision sensors. While the single-robot version of this problem is fairly well understood, much work remains to be done on the multi-robot version. For example, one question is about the effectiveness of multi-robot teams over single-robot versions, and to what extent adding additional robots brings diminishing returns. This issue has begun to be studied, but much remains to be determined for the variety of approaches available for localization, mapping, and exploration.

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 [16]. Each node of an SRT represents a configuration q which was visited by at least one robot, together with the associated local safe region (LSR). 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.

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 $S(q)$. The robots move in a planar workspace, i.e., $\mathbf{R}^2$ 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 [17]. Each robot is a polygon[1] 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 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})
1 $\mathcal{T}.init(q_{init})$
2 BUILD_SRT($q_{init}.\mathcal{T}$);
3 SUPPORT_OTHERS(q_{init});

Fig. 1. The Multi-SRT algorithm.

[1] Polygonal models make it possible to efficiently compute geometric properties, such as areas and visibility regions.

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. For more details of the most important stages of the BUILD_SRT procedure, one can see [16].

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. We have devised two versions of the function, ORGANIZE1 and ORGANIZE2.

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.

```
BUILD_SRT(q_init, T)
q_act = q_init;
do
 BUILD_AND_WAIT_GPR();
 S(q_act) ← PERCEIVE(q_act);
 ADD(T, (q_act, S(q_act)));
 G ←BUILD_GER();
 F(q_act) ← LOCAL_FRONTIER(q_act, S(q_act), T, ∪T_i);
 q_target ← PLANNER(q_act, F(q_act), q_init);
 if q_target ≠ NULL
  if |G| > 1
   (G_f, G_u) ← CHECK_FEASIBILITY(G);
   if G_u ≠ ∅
    q_target ← COORDINATE(G_f, G_u);
 q_act ← MOVE_TO(q_target);
while q_target ≠ NULL
```

Fig. 2. The BUILD_SRT procedure.

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

ORGANIZE1 (organization via arbitration) implements a simple mechanism for arbitration in $\mathcal{G}$. In particular, all the robots contained in the feasible subset $\mathcal{G}_f$ are allowed to move (their target configurations are left unchanged). The robots that are in the unfeasible subset $\mathcal{G}_u$, are not allowed to move (their target configuration is initialized to the current configuration) with the exception of one whose motion is authorized (this strategy is guaranteed to avoid conflicts).

The selection of the robot authorized in $\mathcal{G}_u$ can be done on the basis of several criteria. The one we implemented chooses randomly one of the robots whose local frontier is empty: any of the robots whose target is their parent node (i.e., robots running BUILD_SRT and robots advised to backtrack by the planner) or robots that are moving along the trees initiated by other robots with the goal of helping them in the expansion (robots that are running the SUPPORT_OTHERS procedure and are still in the stage of transfer). This strategy is motivated by the fact that if the motion was unauthorized, such robots will have to wait until their path is clear, because they cannot change their goal (compared to the robots whose local frontier is not empty and the planner can propose a different destination). A non-ethical approach would be to randomly choose between robots in $\mathcal{G}_u$ using a probability proportional to the length of its local frontier.

ORGANIZE2 (organization through replanning) tries to modify the targets of the robots in $\mathcal{G}$, in order to maximize the number of simultaneous feasible moves. This can be done by formalizing the problem as follows. Consider the set of target configurations $\mathcal{Q}_{\mathcal{G}}$ associated with the GER $\mathcal{G}$. Two target configurations in $\mathcal{Q}_{\mathcal{G}}$ are compatible if they can be reached by the corresponding robots with paths that are not intercepted. Let G be the compatibility graph associated with $\{\mathcal{G}, \mathcal{Q}_{\mathcal{G}}\}$ and defined as the indirect graph whose nodes represent the robots in $\mathcal{G}$ and whose arcs connect pairs of nodes with compatible targets. A maximum clique in G is a full subgraph of G with maximum cardinality, corresponding to a maximum subset of robots with compatible targets. The ideal objective of the ORGANIZE2 is to modify the set of target configurations $\mathcal{Q}_{\mathcal{G}}$ to maximize the cardinality of the maximum clique associated with the constraint that the target of each robot is either accepted, changed into another configuration directed to the robot's local frontier (if this is not empty) or to the robot's current configuration (the move is not authorized). This is a very complex problem whose solution requires the computation of maximum cliques as a subproblem. To find a satisfactory solution in a given amount of time, we have adopted a randomized search technique conducted by the master as a sequential game with complete information.

COORDINATE($\mathcal{G}_f, \mathcal{G}_u$)
master_id $\leftarrow$ MASTER_ELECTION($\mathcal{G}$);
if my_id = master_id
  $\mathcal{Q}_{\mathcal{G}} \leftarrow$ ORGANIZE($\mathcal{G}_f, \mathcal{G}_u$);
else
  WAIT;
return $\mathcal{Q}_{\mathcal{G}}$(my_id);

Fig. 3. The coordination function.

The procedure SUPPORT_OTHERS 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.

Research into exploration methods has usually addressed the problem in isolation, rather than together with the SLAM problem. Frontier-based exploration is one of the simplest techniques and has been widely tested. Using this exploration technique, the robot moves towards boundaries or frontiers between known and unknown parts of the environment. Frontier-based exploration is relatively easy to integrate into most existing mobile robots architectures, although its performance in realistic long term robot experiments with uncertainty is untested.

3 Simulation Results

In order to illustrate the behavior of the Multi-SRT exploration approach, we present two strategies, the Multi-SRT-Radial (see [18] for more details.) and the Multi-SRT-Star. The strategies were implemented in Visual C++ V. 6.0, taking advantage of the MSL library's[2] 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 sensor's perception systems[3]. GPC is a C library implementation of a polygon clipping algorithm. In the simulation process, the robot along with the sensor's system move in a 2D world, where the obstacles are static; the only moving object is the robot. The robot's geometric description, the workspace and the obstacles are described with polygons.

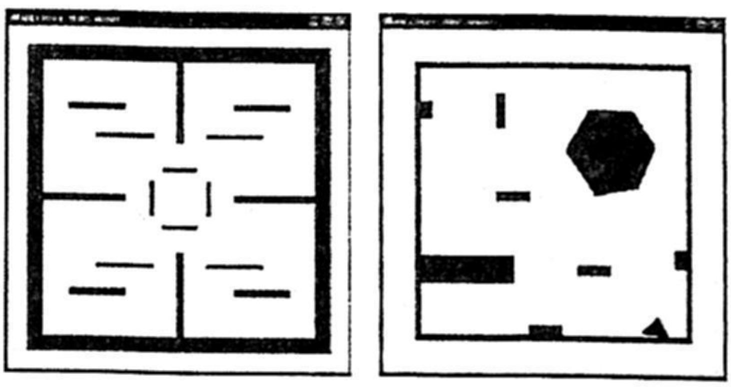

Fig. 4. Environments used for the tests of the Multi-SRT.

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 numerical results were averaged over 20 simulation runs. Environment coverage is not reported because it was complete in all our simulations. Figure 4 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. These first results showed are obtained by using the Multi-SRT-Radial strategy. The performance

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

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

of the method is evaluated in terms of exploration time (the time required by the last robot of the team to return home). The polygonal representation facilitates the use of the GPC library for the perception algorithm's simulation. If S is the zone that the sensor can perceive in absence of obstacles and SR the perceived zone, the SR area is obtained using the difference operation of GPC between S and the polygons that represent the obstacles.

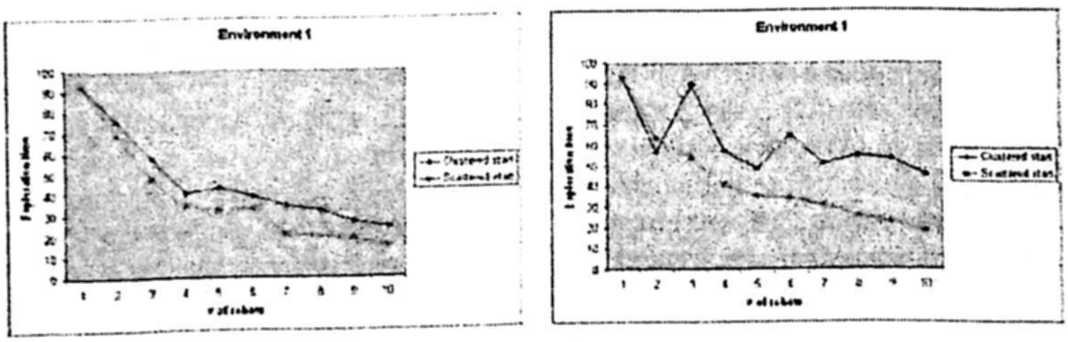

Fig. 5. Environment 1 exploration with scattered and clustered start. To the left with unlimited communication range and in the right with limited communication range.

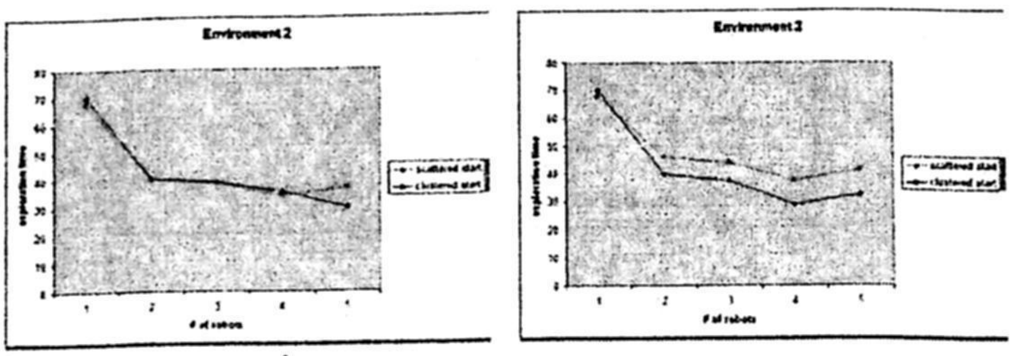

Fig. 6. Environment 2 exploration with scattered and clustered start. To the left with unlimited communication range and in the right with limited communication range.

Exploration time for teams of different cardinality are shown in Figures 5 and 6, 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 (see Figure 7, for the unlimited communication range case). 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. The total travelled distance increases with the number of robots because more robots try to support the others in their expansion. We used ORGANIZE2 for coordination (the performance of ORGANIZE1 is similar). As the number of robots increases, communication chains are formed and the total distance decreases. Due to the corresponding importance of the coordination phase, the waiting time[4] in both coordination strategies grows with the number of robots. We did not report the corresponding results to the exploration using the Multi-SRT-Star strategy because in many cases the exploration of the environment is not completed.

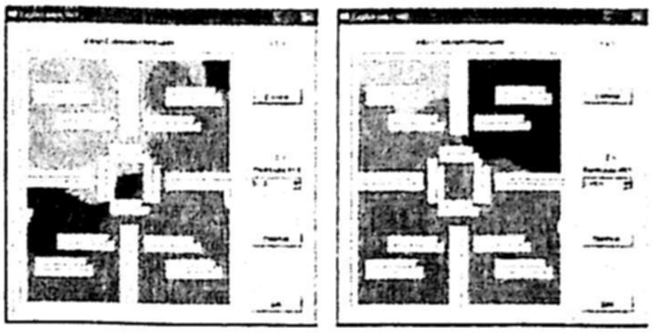

Fig. 7. The explored regions with clustered and scattered starts with a team of 4 robots.

Figures 8 and 9 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. 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.

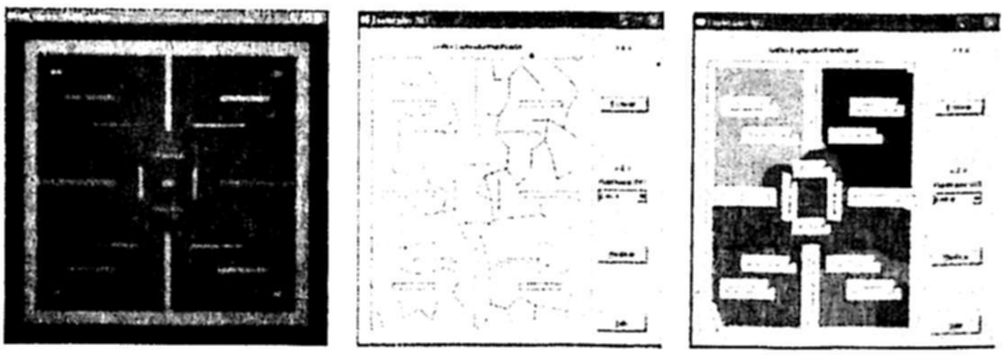

Fig. 8. Environment 1 with 5 robots. The Multi-SRT and explored regions with scattered start. Time = 54.348 secs with 141 nodes.

[4] The average percentage of the exploration time that a robot spends waiting due to coordination.

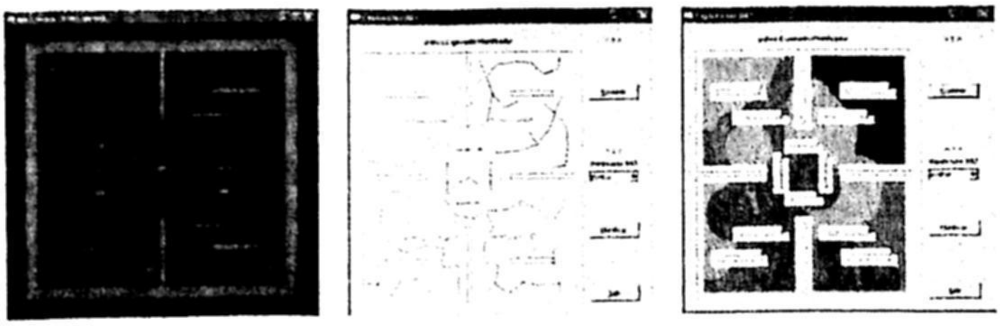

Fig. 9. Environment 1 with 10 robots. The Multi-SRT and explored regions with scattered start. Time = 26.658 secs with 32 nodes.

If we compare the exploration times in the figures 8 and 9, 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 8 and 9, 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.

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.

3.1 Discussion

To solve a multi-agent task, either centralized or decentralized (distributed) approaches can be used. A centralized model uses a powerful agent to plan and schedule the subtasks for every agent. This control agent has a global knowledge concerning the environment and the problems. It can deliberately plan for better performance. However, for tasks with NP complexity, the centralized approach is impractical. Furthermore, the control agent must be powerful enough to achieve satisfactory performance. High design complexity, high cost and low reliability are the other drawbacks of this approach.

On the other hand, a distributed approach decreases design complexity and cost, while increasing the reliability. Agents are autonomous and equal. An agent plans for itself and communicates with the others in order to accomplish the global task. Since every agent interacts directly with the environment, it is reactive. However, each agent has only local knowledge of the task and the environment. Hence, it cannot make the best decision of the global task alone. Furthermore, negotiation and social cooperation rules for conflict resolution are required to coordinate among them.

One distributed approach is to let each agent work alone. Whenever an agent cannot achieve its goal by itself, it requests help from the others. Moreover, every agent always offers help when it can. This is *help-based cooperation.* It is simple and effective in achieving the overall task. Its problems include: too many helpers for an agent is a waste; too many agent requiring help will cause a deadlock; some agents obtain help more often than the other agents (unbalanced load); and local knowledge limits the system performance.

Another approach is to let agents together. For a given task, agents coordinate a global plan for performing the task by exchanging their local knowledge. Its is *coordination-based cooperation.* With enough information, a task can be optimally achieved. Moreover, each agent can optimize its decisions. The problems of this approach include overhead due to coordination, complexity in choosing optimal decisions, and the amount of storage for saving the exchanged information.

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

The use of multi-robot system brings in general many advantages. In exploration, it aims at significantly reducing the time required to complete the task. Possible applications include surface inspection, mine sweeping, surveillance, search and rescue missions and planetary operations. The aim of this research work is to develop a robust exploration technique for single and multi-robot system in an unknown indoor environment. For the case where multiple robots are employed, the system must be scalable, decentralized and tolerant to temporary lost of communications between some robots.

We have presented an interesting approach for cooperative exploration based on the SRT method. 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 the radial perception strategy are more interesting.

SRT is an interesting method for single/multiple robot indoor exploration and mapping. The method combines local frontier-based exploration technique and global graph-based representation of the environment to produce a robust autonomous exploration strategy.

In this proposal. we assumed the robots have good localization and a common frame of reference, but not necessarily the same start position. We are currently working in an integrated exploration strategy for one single robot (the continuous localization procedure is based on natural features of the safe region). The problem of simultaneously localizing a group of mobile robots is still open. 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. E. U, Acar, H. Choset, A. A. Rizzi, P. N. Atkat and D. Hull, "Morse decompositions for coverage tasks", *International Journal of Robotics Research*, Vol. 21, (2002) 331-344
11. M. A. Batalin and G. S. Sukhatme, "Coverage, exploration and deployment by a mobile robot and communication network", *Proc. International Workshop on Information Processing in Sensor Networks*, (2003) 376-391
12. S. Thrun, W. Burgard and D. Fox, "Probabilistic robotics", *MIT Press*, (2005)
13. 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
14. W. Burgard, M. Moors, C. Stachniss and F. Schneider, "Coordinated multi-robot exploration", *IEEE Transactions on Robotics*, Vol. 21, No. 3, (2005) 376-378
15. S. I. Roumeliotis and G. A. Bekey., "Distributed multirobot localization", *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 5, (2002) 781-795
16. A. Toriz Palacios, "Estrategias probabilisticas para la exploración cooperativa de robots móviles", *Master Thesis*, FCC-BUAP (in spanish), 2007
17. I. Kamon, E. Rimon and E. Rivlin "Range-sensor-based navigation in three-dimensional polyhedral environments", *The International Journal of Robotics Research*, Vol. 20, No. 1, (2001) 6-25
18. 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