The Sensor-based Random Graph Method for Cooperative Robot Exploration

Antonio Franchi, *Student Member, IEEE*, Luigi Freda, *Member, IEEE*, Giuseppe Oriolo, *Senior Member, IEEE*, and Marilena Vendittelli, *Member, IEEE*

***Abstract*—We present a decentralized cooperative exploration strategy for a team of mobile robots equipped with range finders. A roadmap of the explored area, with the associate safe region, is built in the form of a sensor-based random graph (SRG). This is expanded by the robots by using a randomized local planner that automatically realizes a tradeoff between information gain and navigation cost. The nodes of the SRG represent view configurations that have been visited by at least one robot, and are connected by arcs that represent safe paths. These paths have been actually traveled by the robots or added to the SRG to improve its connectivity. Decentralized cooperation and coordination mechanisms are used so as to guarantee exploration efficiency and avoid conflicts. Simulations and experiments are presented to show the performance of the proposed technique.**

***Index Terms*—Cooperative exploration, decentralized algorithms, multirobot systems.**

I. Introduction

EXPLORATION of unknown environments is one of the most challenging problems in robotics. This task typically requires a mobile robot to cover an unknown area while learning, at the same time, a model of the environment or locating a given object. A wide range of applications are conceivable, including automated surveillance, search-and-rescue operations, map building, and planetary missions.

Using a multirobot system has a number of potential advantages [1], [2]. A team of robots is expected to be able to complete an exploration task faster than a single robot. Moreover, if a map of the environment is to be built, the redundant information provided by multiple robots can be used to increase the map accuracy and the quality of the localization [3]. To achieve these objectives, some sort of task decomposition and allocation is required. In practice, strategies to conveniently distribute robots over the environment should be devised so as to prevent the occurrence of spatial conflicts [4] and take advantage of the multirobot architecture. Clearly, communication plays a crucial role in achieving cooperative behavior with improved performance [5].

In most exploration strategies, the boundary between known and unknown territory (the *frontier*) is approached in order to maximize the information gain. A pioneering work for the multirobot case is [6]: the robots merge the acquired information in a global gridmap of the environment, from which the frontier is extracted and used to plan individual robot motions. While this basic scheme lacks an arbitration mechanism preventing robots from approaching the same frontier region, it is proposed to negotiate robot targets in [7] by optimizing a utility function that takes into account the information gain of a particular region, the cost of reaching it, and the number of robots currently heading there. The same decentralized frontier-based approach is used in [8], where a large-scale heterogenous team of mobile robots is used for exploration, mapping, deployment, and detection tasks. The utility of a particular frontier region from the viewpoint of relative robot localization is also considered in [9]. In the incremental deployment algorithm of [10], robots approach the frontier while retaining visual contact with each other. An interesting multirobot architecture in which robots are guided through the exploration by a market economy is presented in [11], whereas a centralized approach is proposed in [12], which uses a frontier-based search and a bidding protocol to assign frontier targets to the robots.

The present paper builds on previous research [13], [14] on cooperative robot exploration based on local information only. In particular, the robots of the team cooperatively build a map of the environment in the form of a graph, called *Sensor-based Random Graph* (SRG), which is an evolution of the *Sensor-based Random Tree* (SRT) defined in [15] and [16].

A *node* of the SRG contains a view configuration and the *Local Safe Region* (LSR) perceived from that location; an *arc* between two nodes represents a safe path between the corresponding configurations. The paths stored in the arcs may have been actually traveled by at least one of the robots, or added by joining directly connectable nodes to improve the connectivity of the roadmap. These special arcs are called *bridges* and essentially provide shortcuts.

With respect to [6]–[12], the distinctive aspects of the SRG method are the following.

1) *Complete decentralization*: Each robot independently selects its next destination toward the *local frontier* of its current LSR, thus approaching areas that appear to be unexplored on the basis of all the available information. This simple cooperation strategy is very efficient and automatically achieves a tradeoff between information gain and navigation cost, without the need of defining and optimizing mixed utility functions.
2) *Continuous replanning*: Since the LSR is bounded by the sensor perception range, the next destination of each robot is always nearby. These short-term plans are more secure, less binding, and result in a more flexible decentralized

Manuscript received July 22, 2008; revised November 3, 2008. First published March 27, 2009; current version published April 15, 2009. Recommended by Guest Editor S. Chiaverini.

The authors are with the Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza," 00185 Rome, Italy (e-mail: franchi@dis.uniroma1.it; freda@dis.uniroma1.it; oriolo@dis.uniroma1.it; venditt@dis.uniroma1.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMECH.2009.2013617

task allocation that can quickly adapt to new information becoming available through communication. This also eliminates the necessity of enforcing artificial timeout conditions on the individual task execution.

3) *Guaranteed coordination*: Another consequence of the short span of each robot plan is that coordination is needed to avoid conflicts only when the robots are close to each other. In particular, we are able to explicitly characterize this situation and provide guaranteed coordination strategies. In addition, a lower bound on the communication range that is needed to implement these strategies can be derived.

This paper is organized as follows. Section II lists the assumptions under which the SRG method is presented. The SRG data structure is described in Section III. The architecture of the software implementing the exploration method on each robot is described in Section IV. Central to this architecture is the action planner, described in Section V. The information encoded in the SRG stored in the memory of each robot is updated as explained in Section VI. The robots exchange information according to the communication protocol given in Section VII. In Section VIII, some implementation issues are discussed. Finally, Sections IX and X present simulation and experimental results, respectively, of the proposed strategy.

II. Problem Setting

The cooperative SRG exploration method is presented under the following assumptions.

1) The robots move in a planar workspace $\mathcal{W} \subseteq I\!R^2$.
2) Each robot is a disk of radius ρ, whose configuration q is described by the cartesian position of its center.
3) Each robot is *path controllable*, i.e., it may follow any path in its configuration space with arbitrary accuracy. This assumption is verified for free-flying as well as (most) nonholonomic mobile robots.
4) The robots are equipped with an omnidirectional sensory system that provides the *Local Safe Region* $\mathrm{LSR}(q)$, which is a description of the free space surrounding the robot at q. The LSR is a star-shaped subset of $I\!R^2$, whose maximum radius is bounded by the robot perception range R_p (Fig. 1).
5) Each robot can broadcast the information stored in its memory (or relevant portions of it) within a communication range R_c at any time. The robot identification (ID) number is included in the heading of any transmission. The robot is always open for receiving communication from other robots located inside R_c.

Many of these assumptions are only taken for simplicity and can be relaxed. The assumption of planar workspace is obviously not restrictive: 3-D worlds are perfectly admissible as long as the sensory system allows the reconstruction of a *planar* LSR for planning the robot motion. Assumption 2 implies that the configuration space of each robot is a copy of the workspace with the obstacles grown so as to allow for the robot size [17]. This assumption is only taken for ease of presentation: the proposed method is readily applicable to robots with arbitrary shape. In Assumption 3, path controllability can be replaced with (simple) controllability provided that a *regional* path planner (i.e., an algorithm that generates feasible paths in a limited region) is available. Assumption 4, and in particular the star-shaped hypothesis, is consistent with the physics of the most common proximity sensors, i.e, range finders, but it also applies to more sophisticated perception techniques (e.g., panoramic vision).

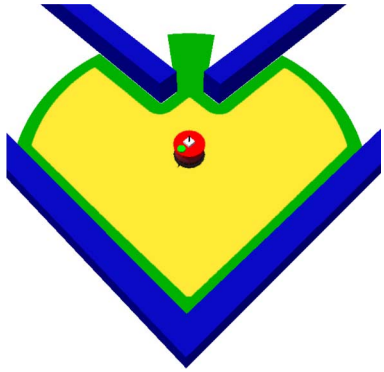

Fig. 1. Lightly colored (green/yellow) area is the LSR at the current configuration. Its lighter (yellow) subset is the LRR. Obstacles are darkly colored (blue).

At this stage, our exploration task can be informally defined as follows: the objective is *to cooperatively cover the largest possible portion of the workspace with sensor perceptions*. A more formal definition will be given in the following in connection with the termination condition for our method.

III. The Sensor-based Random Graph (SRG)

The SRG is a compact data structure used to represent the area explored by the team of robots as well as with the history of the exploration process in the form of a roadmap. Each node of the SRG represents a collision-free configuration q that has been visited by at least one robot. The result of the perception process at q is the associated $\mathrm{LSR}(q)$. An arc between two nodes represents a collision-free path between the two configurations. This path may have been actually traveled by at least one robot, or added to the SRG, to improve its connectivity (in this case, it is called *bridge*, see Section VI).

It should be emphasized that the SRG is a "virtual" data structure, whose knowledge is *distributed* in the team [18]. The ith robot knows its own version SRG_i of the graph. SRG_i is built by the robot on the basis of data acquired either by the robot itself or via communication with other robots. The SRG, which is the union of all the SRG_i's, is not explicitly represented at any level.

Each robot incrementally builds its own SRG_i by extending it in the most promising direction via a biased random mechanism. In doing this, it uses a local coordination strategy that takes into account the information coming from other robots in order to guarantee that the distributed knowledge is increased. As a result, the SRG as a whole is simultaneously extended in several directions toward currently unexplored areas.

For planning robot motions, the SRG method makes use of three structures that are directly derived from the LSR: the *Local Reachable Region* $\mathrm{LRR}(q)$, the *Local Frontier* $\mathrm{LF}(q)$, and the *Local Informative Region* $\mathrm{LIR}(q)$.

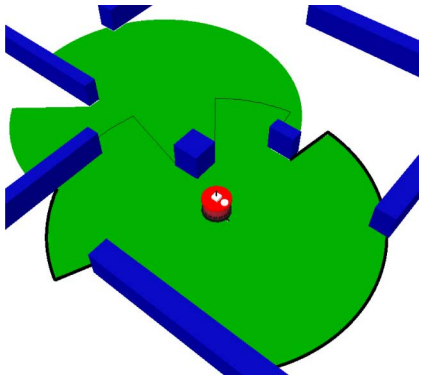

Fig. 2. Frontier arcs (thick lines) and free arcs (thin lines) of the current LSR boundary.

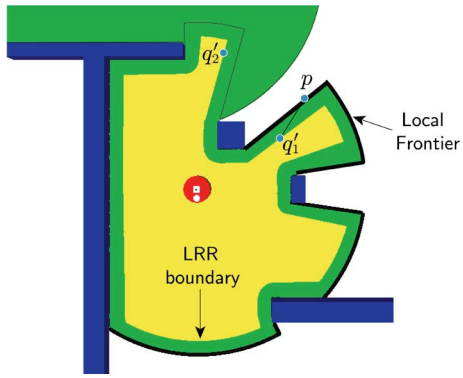

Fig. 3. While q_1' is a local informative configuration, q_2' is not.

A. Local Reachable Region (LRR)

In the SRG method, the action planning domain is the robot configuration space and, in particular, the *Local Reachable Region* $\mathrm{LRR}(q)$, defined as the set of configurations that can be reached from q with the robot staying in $\mathrm{LSR}(q)$ (see Fig. 1). Under Assumptions 2 and 3, the $\mathrm{LRR}(q)$ can be obtained by *eroding* the $\mathrm{LSR}(q)$ with the robot disk as structuring element [19], and then *extracting* the connected component containing q. The LRR is not star-shaped in general.

B. Local Frontier (LF)

To identify promising exploration actions from the available information, the robot identifies the portion of the current LSR boundary leading to unexplored areas. To this end, the boundary $\partial\mathrm{LSR}$ is partitioned in *obstacle*, *free*, and *frontier* arcs (see Fig. 2). An *obstacle arc* is a portion of the boundary of the obstacle region as detected from q. Under Assumption 4, these are reconstructed from the range scan by identifying contiguous readings that are smaller than the perception range R_p. Points of $\partial\mathrm{LSR}$ that fall in other LSRs stored in the SRG_i belong to *free arcs*. Any arc that is neither obstacle nor free is a *frontier arc*, and by construction, identifies the transition from explored to unexplored regions. The union of the frontier arcs of $\mathrm{LSR}(q)$ is the *Local Frontier* $\mathrm{LF}(q)$.

A frontier arc of $\mathrm{LSR}(q)$ is computed from the whole SRG_i, which, in turn, is built using all the information available to the robot. This is the key to our decentralized cooperation mechanism aimed at optimizing the exploration performance of the team. Such a mechanism is inherently local and contingent because it relies on communication between the robots.

C. Local Informative Region (LIR)

Given an $\mathrm{LSR}(q)$ and its $\mathrm{LRR}(q)$, a $q' \in \partial\mathrm{LRR}(q)$ is called a *local informative configuration* if there exists a point p on $\mathrm{LF}(q)$ such that (see Fig. 3):

1) the open line segment (also called the *line of sight*) joining p and q' does not intersect the boundary $\partial\mathrm{LSR}(q)$;
2) $\|p - q'\| < R_p$.

The first condition guarantees that $p \in \mathrm{LF}(q)$ is "visible" from q' through a line of sight contained in $\mathrm{LSR}(q)$, while the second ensures that p is contained in the perception range at q'. Together, they guarantee that a sensor scan taken from q' will "push forward" the frontier arc containing p, thereby increasing the information about the explored workspace.

Conditions 1 and 2 may be satisfied also at configurations that belong to the interior of LRR. However, the aforesaid definition, according to which local informative configurations must be on the LRR boundary, aims at maximizing the information gain. A local informative configuration has positive *information gain* in the sense of [20] and [21].

The *Local Informative Region* $\mathrm{LIR}(q)$ is the set of all local informative configurations; $\mathrm{LF}(q) = \emptyset$ implies $\mathrm{LIR}(q) = \emptyset$.

IV. Functional Architecture

For the sake of clarity, the SRG exploration will be described with reference to the functional architecture of the software running on each robot of the team, shown in Fig. 4. Blocks with thick edges represent processes, those with thin edges represent threads, and dashed rectangles represent data. Arrows indicate information flow: thick for interprocess communication, thin for communication between threads, and dashed for read/write operation on data structures.

The *robot explorer* implements the SRG exploration algorithm, while the *robot driver* provides low-level primitives for motion, localization, and perception (not discussed in this paper). The two processes communicate through the transmission control protocol (TCP), allowing a distributed instantiation of the architecture and providing a flexible integrated environment for simulation and experimental validation. With this architecture, in fact, the explorer and the driver do not need to run on the same machine, and the latter can be a real or a simulated robot.

The robot explorer is realized by four threads: the *action planner*, the *SRG manager*, the *broadcaster* and the *listener*. The action planner is the core of the robot explorer: it is in charge of choosing the next exploration action in a cooperative and coordinated way. The task of the SRG manager is to elaborate and continuously update the data stored in the SRG_i on the basis of the information received from the action planner or, through the listener, from the rest of the team. In particular, the action planner makes available the LSRs acquired by the robot,

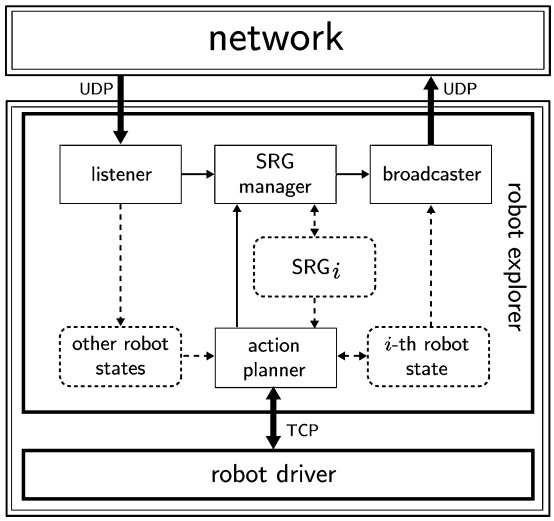

Fig. 4. Functional architecture of the software implementing the SRG method on the ith robot.

```
ACTION PLANNER
1:   build GPA and synchronize
2:   perceive LSR
3:   build GEA
4:   if LIR is non-empty
5:      select a target configuration on LIR and
        plan a path leading to it in the LRR
6:   else if there exists a node of SRG_i with non-empty LIR
7:         find the the closest node with non-empty LIR and
           plan a path on SRG_i leading to it
8:         set as target the first adjacent node on the path
9:      else
10:        terminate exploration: homing
11:  if |GEA| > 1
12:     check feasibility of the GEA robot paths
13:     if there are unfeasible paths
14:        choose a master in the GEA
15:        wait target from master
16:  issue 'move to target' command
```

Fig. 5. Pseudocode description of the action planner.

while the listener provides the SRG_j's built by other robots of the team with which communication has taken place. The SRG manager incorporates these data so as to maintain the consistency of local representations. To this end, self-localization and mutual localization information coming from the robot driver (and, through the listener, from other robots) are used. Finally, the broadcaster transmits all the information currently available to the robot.

V. Action Planner

The action planner for the ith robot is described in pseudocode in Fig. 5. Its basic steps are first briefly discussed, and then detailed in the rest of the section.

At the beginning, the robot is stationary in a position corresponding to a node of the SRG_i. The first operation that the robot performs is the identification of the *Group of Engaged Agents* (GEA), i.e, the other agents of the team with which cooperation and coordination are necessary. This is achieved by first building the *Group of Preengaged Agents* (GPA), i.e., the robots that are candidates of the GEA, and synchronizing with them (line 1). These computations are performed by the robot on the basis of the information stored in its SRG_i and data coming from other robots (see Fig. 4). Once synchronization has been achieved, the action planner sends a "perceive" command to the robot driver, receives from it the current LSR, and makes it available to the SRG manager (line 2). When the perceptions of all the robots in the GPA have been received, the actual GEA can be built using simple geometry. Lines 1–3 are detailed in Section V-A.

If the LIR (computed by the SRG manager) of the current LSR is nonempty, the action planner selects a *target* configuration (also called *view* configuration in the following) on the LIR according to a randomized mechanism (line 5). A path reaching the target is then planned inside the current LRR (this guarantees that the path is safe, i.e., collision-free on the basis of the available knowledge). If the current LIR is empty, the action planner verifies whether there exists in SRG_i a node with nonempty LIR (line 6). In the positive case, it first finds the closest node with a nonempty LIR and plans a path leading to it on SRG_i (line 7). Then, it selects as target the first adjacent node on such path (line 8). See Section V-B for a commentary of lines 5–8.

After the target is selected, the robot checks if its GEA includes other robots. In the negative case, the robot directly moves to its target. Otherwise, the prospective paths of the robots in the GEA are checked for mutual collisions, and accordingly, classified into feasible and unfeasible paths (line 12, Section V-C). If there are unfeasible paths, a GEA coordination phase takes place. A master robot is selected in the GEA, which may either confirm or modify (see Section V-D) the current target of the robot. In particular, the robot move may be simply forbidden by resetting the target to its current configuration. Last, the target is received from the master (line 15) and the robot moves toward it (line 16).

The action planner terminates the exploration process when the condition of line 9 is verified, i.e., when the LIRs of all nodes in the SRG_i are empty (hence, no local informative configurations remain in SRG_i). At this point, the robot enters a *homing* phase, in which it plans and follows a path on the SRG_i leading back to its starting configuration.

If all nodes of the SRG_i have empty LF, the aforementioned termination condition is obviously met. However, such a condition may also be satisfied in the presence of nonempty LFs, as in the case of Fig. 6. Here, there is no further exploration action that will allow the robot to move/remove the LF (a typical situation when the workspace contains "windows" across which the robot can "see" but cannot "pass").

It is also interesting to point out that, although our exploration task is cooperative in nature, the aforementioned termination condition is consistent with a decentralized approach, as it can be computed by each robot on its own. In fact, when no local informative configurations are left in its SRG_i, a robot can safely exit the exploration process, as it will never be able to contribute new perceived areas to the distributed SRG. Actually, our simulations and experiments have shown that, on the

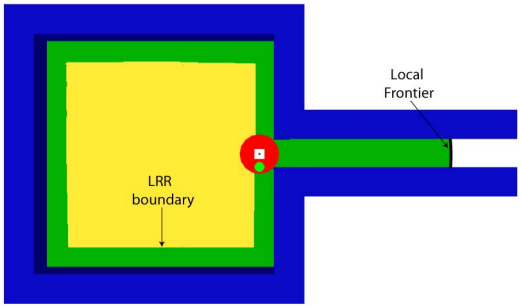

Fig. 6. Typical situation in which the LIR is empty while the LF is not. Once the robot is in contact with the narrow passage boundary, no further exploration action will allow to move/remove the LF (the robot can "see" but cannot "pass" through the opening).

average, all robots tend to perform homing simultaneously, thereby supporting a claim of efficient exploration.

A. GPA/GEA Construction

At the start of the action planner algorithm, the robot is stationary and needs to identify other robots whose LSRs may overlap with its own, in order to *cooperate* (avoid inefficient actions) and *coordinate* (avoid collisions) with them. The other robots may be stationary as well (in this case, their targets coincide with the current configuration) or moving toward a target; hence, a synchronization phase is needed.

Two robots are said to be *GPA-coupled* if the distance between their targets is at most $2R_p$, i.e., twice the perception range. The GPA of the robot is then built by grouping together all the robots to which it can be connected through a chain of GPA couplings (see Fig. 7, top). To achieve synchronization, the GPA is computed and updated until all its members are stationary; when this is achieved, the robot exits from this synchronization phase. If T is the maximum time required to reach the target, the upper bound of the waiting time is $(N-1)T$, where N is the number of robots of the team. T is bounded; in fact, the target configuration is at a distance at most $R_p - \rho$, since it is always in the current LIR.

The communication range R_c clearly plays a role in the GPA construction. Since the maximum distance between the robot and any other robot with which it is GPA-coupled is $3R_p - \rho$ (the other robot may still be moving to its target, which, however, cannot be farther than $R_p - \rho$ from the current configuration), it is sufficient to assume $R_c \geq 3R_p - \rho$ to guarantee that the GPA accounts for all the robots that are candidates to the GEA.

Once the robot is synchronized with its GPA (and all the LSRs of the other robots in the GPA are available), it builds the GEA, i.e., the robots with which cooperation and coordination are actually necessary. If we define two robots to be *GEA-coupled* when their LSRs overlap, the GEA of the robot (see Fig. 7, bottom) comprises all GPA robots to which it can be connected through a chain of GEA couplings. Synchronization guarantees that all the GPA robots are stationary when the GEA is computed. The GEA is *symmetric*, i.e., it is the same for all robots in the group.

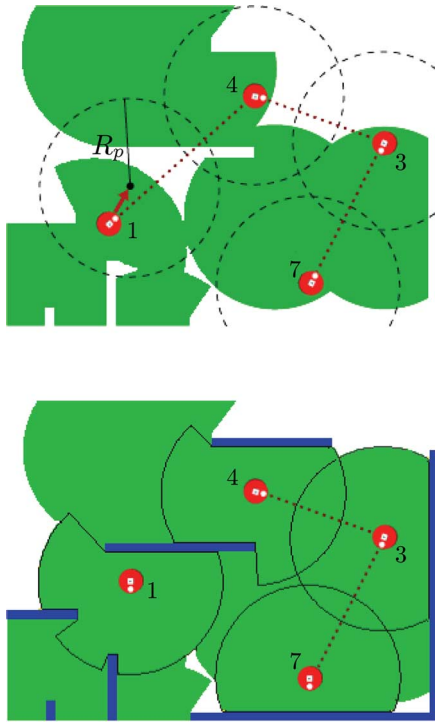

Fig. 7. Example of GPA/GEA construction. *Top*: The GPA of robot 4 consists of robots 1, 3, 4, and 7: robot 1 is still moving toward its target point, while robots 3, 4, and 7 are stationary. The perception areas of the robots (prospective in the case of robot 1) overlap in pairs. *Bottom*: Once the LSR have been computed, only robots 3, 4, and 7 belong to the GEA of robot 4 since their LSRs overlap in pairs.

The GEA is a cornerstone of our method, as it identifies a group of robots that, in view of their vicinity, spontaneously agree to cooperate and coordinate with each other on a temporary basis. Such an agreement can be reached with a limited communication range ($R_c \geq 3R_p - \rho$).

B. Target and Path Generation

If the current LIR is nonempty, the action planner chooses a target configuration in the LIR using a randomized mechanism.

In particular, the LIR is, in general, the union of disjoint arcs $a_1, a_2, \ldots, a_m$. First, one of these arcs is selected with a probability proportional to its length. Let a be the selected arc and s be the arc length parameter along a, with $s \in [0, L]$. The target configuration is selected on a by generating a random value s^* according to a normal distribution with mean value $L/2$ and standard deviation $\sigma = L/6$, and taking the configuration identified by $s = s^*$. At this point, a path to the target can be planned in the current LRR; note that, in view of Assumption 3, any such path can be executed by the robot (and is collision-free by definition).

A deterministic version of this target selection procedure can be envisaged, in which a specific configuration in the LIR is selected according to some fixed criterion, e.g., maximum information gain. However, our experience indicates that using this strategy the computational load is four or five times worse than our randomized version, which is not justified by the

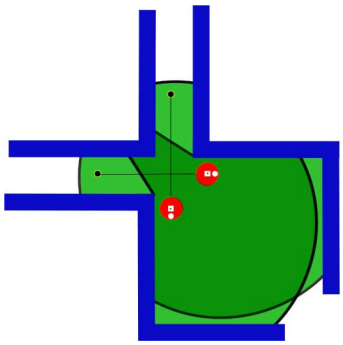

Fig. 8. Prospective paths of robots belonging to the GEA may intersect as each of them tries to move toward its LF.

improvement in performance (traveled distance and number of views), which is lower than 5% on the average.

Due to the GPA synchronization phase, all the robots in a GEA plan at the same time, and therefore, the cooperation mechanism encoded in the notion of LF is enforced throughout the group. This "agreement of intents" is realized without any centralized decision module.

C. GEA Path Feasibility Check

Although the current LF of a robot cannot belong to the LSR of another robot of the GEA (see Fig. 2), the two prospective paths may still intersect (see Fig. 8).

Let $\mathcal{G}$ be the set of robots in the GEA. The prospective paths of the robots of $\mathcal{G}$ are checked to establish whether they are simultaneously feasible, i.e., they do not lead to collisions (for simplicity, the possibility of velocity scaling along the paths is not considered). All pairs of paths that intersect are identified, and the corresponding robots stored in the GEA *unfeasible subset* $\mathcal{G}_u$. The remaining robots are the GEA *feasible subset* $\mathcal{G}_f$. The complexity of this check is $O(|\mathcal{G}|^2)$.

D. Coordination

If the subset $\mathcal{G}_u$ of robots with unfeasible paths is nonempty, a coordination phase takes place locally. At first, a *master robot* within $\mathcal{G}$ is elected.[1] This can be accomplished in many ways through a deterministic procedure known by all the robots; for instance, the robot with the higher ID number can be chosen. Two cases are then possible.

1) If the robot is the master, it builds the vector of targets $\mathcal{Q}_\mathcal{G}$ that collects the target configurations received from the GEA robots. Then, it rearranges this vector so as to obtain a feasible collective motion. Here, rearrange may mean either simply accepting/resetting the target of a robot to the current configuration (i.e, authorizing/forbidding the move) or adding a third option, i.e., changing it to a new target. Correspondingly, we have devised two strategies, i.e., *coordination via arbitration* and *coordination via replanning*(see next).
2) If the robot is not the master, it waits for until the receipt of a specific signal from the master.

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

1) Coordination via Arbitration: This strategy implements a simple arbitration mechanism on $\mathcal{G}$. All the robots contained in the feasible subset $\mathcal{G}_f$ are allowed to move (their target configuration is left unchanged). The robots in the unfeasible subset $\mathcal{G}_u$ are not allowed to move (their target is reset to the current configuration) with the exception of a single one whose motion is authorized (by construction, this strategy is guaranteed not to produce conflicts).

The selection of the authorized robot in $\mathcal{G}_u$ may be done on the basis of various criteria. The one we have used chooses randomly one of the robots (if any) whose LIR is empty. This strategy is motivated by the fact that, if their move is not authorized, such robots will have to wait for their path to become clear, as they cannot change their target (as opposed to robots whose LIR is nonempty, to which the random planner may propose a different target in the next cycle). An antithetical criterion would be to use a probability proportional to the LIR extension to choose randomly a robot in $\mathcal{G}_u$.

2) Coordination via Replanning: This strategy tries to modify the targets of the robots in $\mathcal{G}$ so as to maximize the number of simultaneous feasible moves. This may be done by formalizing the problem as follows.

Consider the set of targets $\mathcal{Q}_\mathcal{G}$ associated to the GEA $\mathcal{G}$. Two targets in $\mathcal{Q}_\mathcal{G}$ are called *compatible* if they can be reached by the corresponding robots with paths that do not intersect. Let G be the *compatibility graph* associated to $(\mathcal{G}, \mathcal{Q}_\mathcal{G})$ and defined as the indirect graph whose nodes represent the robots in $\mathcal{G}$ and whose arcs join pairs of nodes with compatible targets. A *maximum clique* of G is a complete subgraph of G with maximum cardinality, corresponding to a maximum subset of robots with compatible targets. The identification of a *maximum clique* is a well-known NP-complete problem in the context of the graph theory [22], [23].

The ideal objective of the replanning strategy is to modify the set of targets $\mathcal{Q}_\mathcal{G}$ so as to maximize the cardinality of the associated maximum clique(s), with the constraint that the target of each robot is either accepted or changed to another configuration on the LIR (if this is nonempty) or to the current robot configuration (the move is not authorized). This is a very complex problem whose solution would require 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, performed by the master as a sequential game with complete information.

VI. SRG Manager

The SRG manager updates SRG_i on the basis of the new information it receives, which consists of one or more nodes to be added to the graph. A node is a view configuration and comes with the associated LSR, a list of adjacent nodes, and the

[1] A master robot is not actually needed when a deterministic coordination algorithm is chosen; in fact, in this case, each robot can run the algorithm on its own, reaching the same solution as the others.

corresponding arcs. Information may reach the SRG manager via two different routes (see Fig. 4). Views gathered by the ith robot itself come from the robot driver through the action planner whereas those collected by other robots are received through the listener. Since each robot uses its own reference frame, views arriving via the second route must undergo a preliminary roto-translation, which is computed on the basis of mutual localization data.

For each new node, the SRG manager:

1) adds the node to SRG_i or, if it is already present, updates its LSR;
2) computes the LRR, the LF, and the LIR;
3) updates the LF and the LIR of the nodes in SRG_i whose LSRs have a nonempty intersection with $\text{LSR}(q)$.

After updating the SRG_i, an important operation is performed aimed at improving the connectivity of the graph. In particular, for each new node v that has been added, the SRG manager identifies all nodes w of SRG_i that satisfy the following conditions:

1) the *graph distance* between v and w is greater than a certain threshold (typically, a small multiple of R_p);
2) $\text{LRR}(v) \cap \text{LRR}(w) \neq \emptyset$.

A *bridge* is then created for each pair (v, w) by planning a path joining the two nodes; note that a safe path between them certainly exists in view of the second condition. The first condition guarantees that only significant improvements to the graph connectivity are enforced; this avoids an excessive increase of the graph complexity.

Once a bridge has been created, it may be mapped to the SRG_i in two different ways, depending on the euclidean distance between v and w, which is bounded by $2(R_p - \rho)$. If $\|v - w\| < R_p - \rho$, the bridge directly becomes an arc. Otherwise, it is encoded as an arc–node–arc sequence, with the intermediate node placed inside $\text{LRR}(v) \cap \text{LRR}(w)$. In this way, we preserve the property that the distance between two adjacent nodes on the graph is bounded by $R_p - \rho$ (which we have used, for example, in Section V-A).

At the time of its creation, a bridge has not been crossed by any robot of the team. Similarly, when a bridge is encoded as an arc–node–arc sequence, the intermediate node has not yet been visited by any robot.

Bridges essentially represent shortcuts that are very useful for performing an efficient exploration. Their addition is particularly relevant in view of the specific nature of the basic SRG method, in which arcs would be created only when the robot is moving toward a new view configuration. If no such configuration exists, the robot is forced to move on the current SRG_i, whose connectivity hence plays a role in the process. See the example of Fig. 9 for illustration.

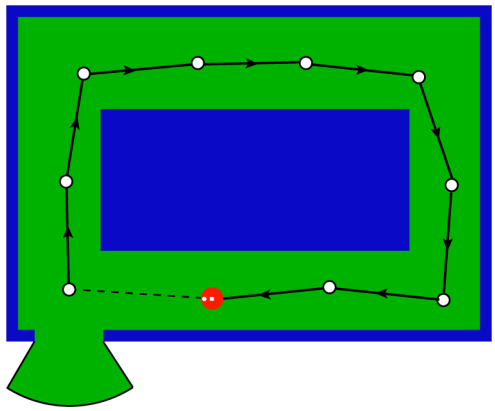

Fig. 9. The importance of adding bridges should be clear from this example. Empty circles represent SRG nodes, while solid line segments are SRG arcs. Without the bridge (which is shown as a dashed line segment) the robot would have to trace back its path around the central obstacle in order to exit the room and continue the exploration.

VII. Broadcaster and Listener

On each robot, the broadcaster and the listener, respectively, transmit and receive information. Conceptually, such information is organized in two possible messages:

1) the *robot state*, i.e., its current configuration, target, GPA/GEA and step of the action planner being executed;
2) the current SRG_i.

Each robot broadcasts its state on a regular basis, whereas its SRG data structure is only transmitted as new data is made available by the SRG manager. The listener receives asynchronous messages from the network and makes them available to the action planner (as other robot states) and the SRG manager. See again Fig. 4.

VIII. Implementation Issues

Before presenting simulation and experiment results, we provide in this section some details about our implementation of the SRG method.

Each LSR is stored in the form of an array of range readings, as returned by the robot range finder. Such an array is a discrete representation of the polar function describing the LSR boundary. The corresponding LF can then be extracted as described in [16].

The LRR can be efficiently built if a gridmap is used to represent the LSR (or the whole workspace). In this case, the LRR at q can be computed as follows:

1) represent the boundary and the interior of the LSR as occupied cells and empty cells, respectively;
2) apply an euclidean distance transform [24] to identify the set E of empty cells whose distance to the closest occupied cell is larger than ρ (the robot radius);
3) compute the LRR as the connected component of E containing q.

Once the LRR is computed, the LIR can be obtained via a *ray casting* procedure. In particular, for each cell representing a configuration q' on the LRR boundary, the LF is inspected until a point p is found (if it exists), which satisfies the two conditions identifying local informative configurations (see Section III-C). Heuristics can be used to speed up the search for such points at contiguous configurations.

In the implementation of the SRG method, it is also necessary for each robot to detect and remove *occlusions*, i.e., obstacle

arcs that are caused by other robots rather than by environment obstacles. Candidate occlusions are identified directly from the range scan profile as protrusions of compatible size. They are then validated using the mutual localization method proposed in [25]: occlusions that are attributed to actual robots are removed and reclassified as frontier arcs.

Other relevant implementation issues concern data transmission among the robots. First, we emphasize that the transmission of GPA/GEA information is necessary (and, hence, performed) only if the communication range R_c is limited. Second, consider that the limitation of R_c does not mean that the cooperation and coordination area is accordingly limited—robots belonging to the same GPA/GEA may be farther than the communication range. In this case, a chain of communication is established to propagate the information between robots that are not in direct communication.

Moreover, in the broadcaster thread, it is not necessary to broadcast the whole SRG_i whenever it is modified. Simple strategies may be used to minimize the amount of transmitted information; for example, timestamps can be used by a robot to identity the portion of data received thus far. In this case, specific peer-to-peer communication strategies can be used to transmit and receive only new information.

IX. Simulations

We present some simulation results to evaluate the performance of the SRG method. To assess the effectiveness of the notion of "bridge," we have also implemented a version that does not add such structures; such version is referred to as SRT in the following, and essentially corresponds to the method described in [13]. The simulations are performed in Move3D [26], a software platform developed at LAAS-CNRS and dedicated to motion planning. The exploration team comprises a varying number of MagellanPro robots. Each robot has a diameter of 0.40 m and carries a 360° laser range finder, with a perception range of 1.60 m. The communication range is set to its minimum admissible value, i.e., $R_c = 3R_p - \rho = 4.60$ m. Two nodes are candidates to be connected by a bridge if their graph distance is at least $3R_p$. Coordination is achieved via replanning.

The performance of the methods is evaluated in terms of *exploration time* (the time required by the last robot of the team to return home) and *traveled distance* (the average distance traveled by each robot). These values are, respectively, expressed as a percentage of the values obtained with a team comprising a single robot using an SRT method. Environment coverage is not reported because it was complete in all cases. In view of the randomized nature of our method, numerical results for each scenario are averaged over ten simulation runs.

The first two groups of simulations are performed in the same garden-like environment, which is a square area with a side of 17 m, and refer to different initial deployments of the team. In the first, the robots are initially scattered in the environment (as if they had been parachuted). In the second, more realistic for environments with a single main entrance, the exploration is started with the robots grouped in a cluster.

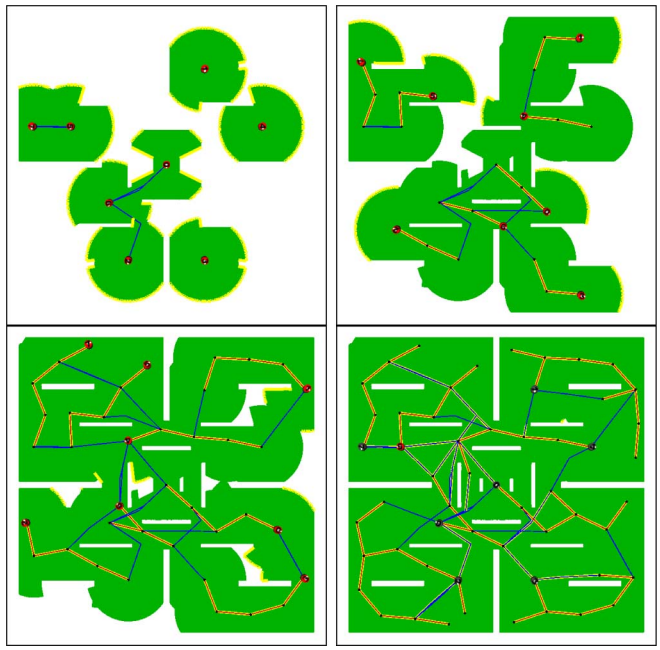

Fig. 10. Simulation 1: SRG garden exploration with scattered start.

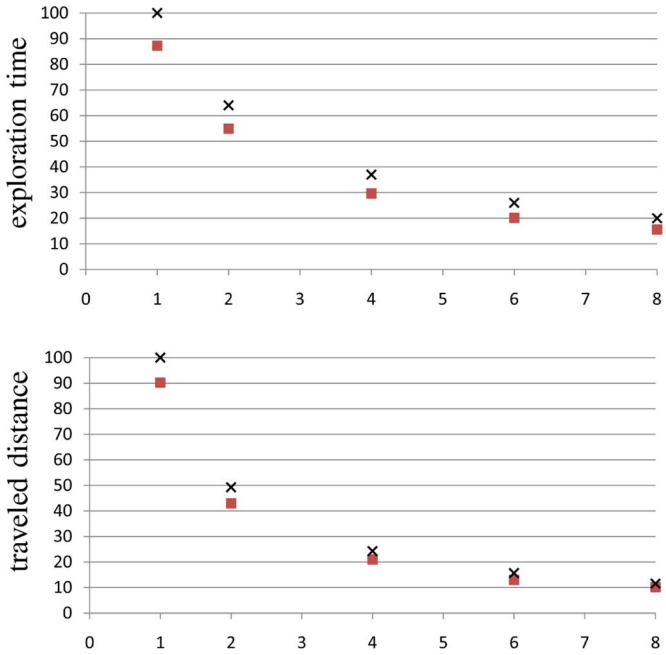

Fig. 11. Garden exploration with scattered start: exploration time (above) and traveled distance (below) with teams of different cardinality. Results for SRG (squares) and SRT (crosses) explorations are shown.

Fig. 10 shows the progress[2] of a typical SRG exploration with a team of eight robots and a scattered initial deployment. Green areas represent the region so far explored. The robots

[2]Video clips of all simulations and experiments are available at the Web page http://www.dis.uniroma1.it/∼labrob/research/multiSRG.html

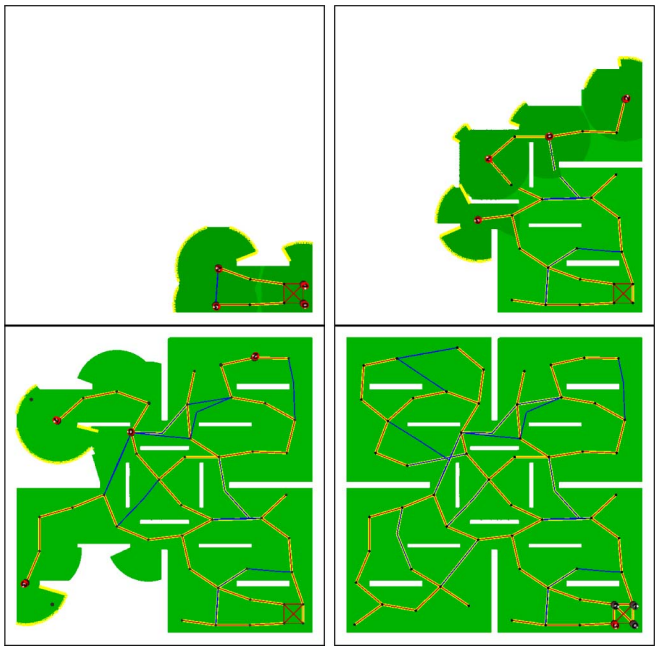

Fig. 12. Simulation 2: SRG garden exploration with clustered start.

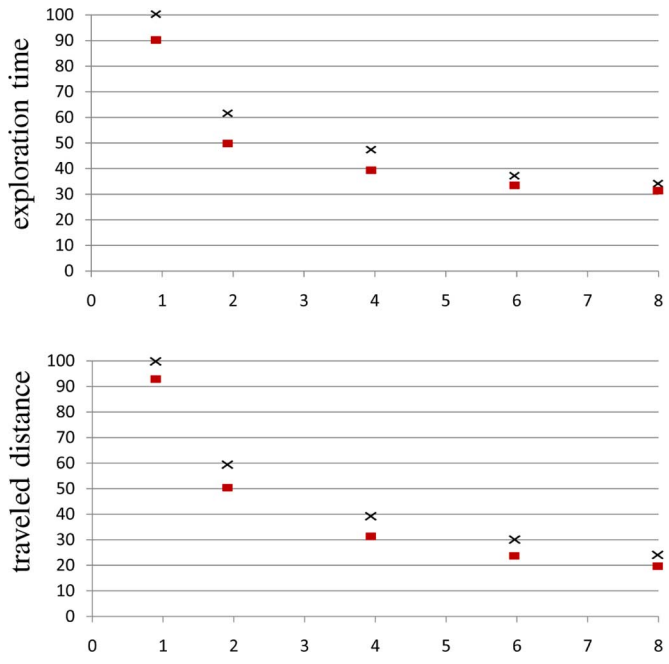

Fig. 13. Garden exploration with clustered start: exploration time (above) and traveled distance (below) with teams of different cardinality. Results for SRG (squares) and SRT (crosses) explorations are shown.

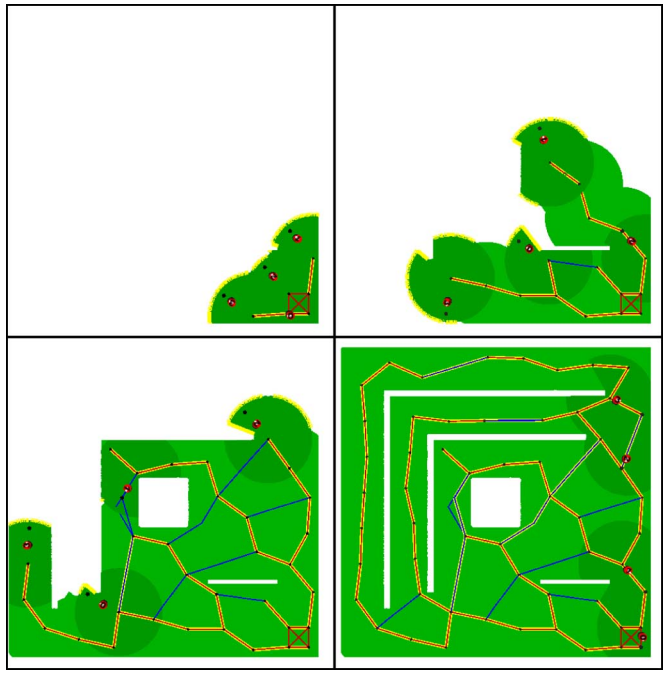

Fig. 14. Simulation 3: SRG corridor exploration with clustered start.

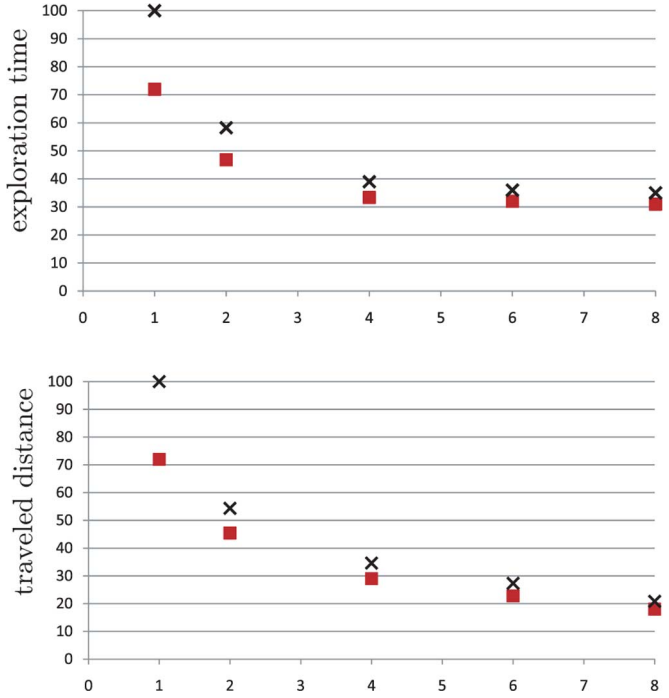

Fig. 15. Corridor exploration with clustered start: exploration time (above) and traveled distance (below) with teams of different cardinality. Results for SRG (squares) and SRT (crosses) explorations are shown.

are represented by red circles with the ID number. The view configurations are marked by black points. Yellow segments represent paths traveled by the robots during the exploration. Bridges are depicted in blue and may or may not have been traversed by the robots. Exploration time and traveled distance for teams of different cardinality are shown in Fig. 11. Average results are shown for both the SRG (squares) and the SRT (crosses) method; in all simulations, variance for these data was less than 5%. As the number of robots in the team increases, the exploration time quickly decreases and tends asymptotically to zero (consider that an increment in the number of evenly deployed robots corresponds to a decrement of the individual areas they must cover, until no motion at all is necessary). A similar

behavior is observed for the traveled distance. The performances of the two methods for a scattered start tend to become similar as the number of robots increases. This is due to the scattered initial deployment, which leads each robot to perform most of the exploration "on its own." As a result, the bridges present in SRG are rarely traversed and the performance of the SRG method does not improve significantly over the SRT method.

Fig. 12 shows the progress of another SRG exploration in the garden-like environment, now with a team of four robots and a clustered initial deployment. Fig. 13 summarizes the performance of the SRG and SRT methods for teams of different cardinality. In this case, the exploration time asymptotically tends to a nonzero value, which approximately represents the time required by a single robot to perform a roundtrip between the cluster center and the farthest point in the environment. Instead, the average distance traveled by each robot still tends to zero. The results show that, on the average, the SRG method introduces a significant improvement in both the exploration time and the traveled distance.

This improvement becomes even more evident in the corridor environment (same size as the garden) used in the third group of simulations. Fig. 14 shows the typical progress of an SRG exploration obtained with a team of four robots and a clustered initial deployment. From the numerical results in Fig. 15, obtained considering teams of different cardinality, it is clear that the marginal utility of an increase in the number of robots is higher for teams of small cardinality.

To investigate the influence of the communication range on the performance of the SRG method, we have repeated the first simulation with a team of eight robots for increasing values of R_c (all satisfying the condition $R_c \geq 3R_p - \rho$). The results, shown in Fig. 16, indicate that a moderate improvement is obtained both in terms of exploration time and traveled distance. As R_c becomes comparable to the size of the square environment, however, an all-to-all communication condition is approached and the marginal utility of an increase in R_c tends to zero.

The cost associated to our coordination mechanism can be quantified by considering that in all our simulations the average cardinality of the GPA/GEA was lower than 2, and the percentage of exploration time spent by each robot in a waiting mode was around 15%.

X. Experiments

The SRG method has been experimentally validated using a team of Khepera III robots.

A. Description of the Robots

Khepera III is the latest release of a family of two-wheel differentially driven mobile minirobots developed by K-TEAM Corporation. The chassis of the robot is 7 cm high and 13 cm wide. It contains two motors, transmission elements, electronics, and a battery pack. A passive caster provides static stability to the vehicle. Each wheel is driven by a DC brushed servomotor coupled to the wheel via a 43.2:1 reduction. An embedded incremental encoder, placed on the motor axis, gives 16 pulses

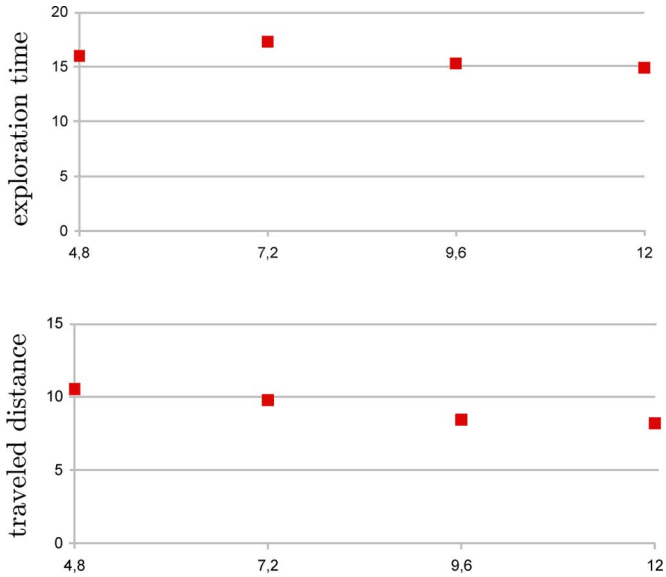

Fig. 16. Garden exploration with a team of eight robots and a clustered start: SRG exploration time (above) and traveled distance (below) for different values of the communication range R_c.

per revolution of the motor. Considering that the diameter of each wheel is equal to 4.1 cm, this results in a resolution of 691 pulses per revolution of the wheel, that correspond to 54 pulses per 0.1 cm of robot motion. The encoder resolution is by default set to the mode $4\times$, which corresponds to 2764 measures per wheel revolution.

In addition to the standard suite (infrared and ultrasonic sensors, serial and universal serial bus (USB) communication), each robot has been equipped with a WiFi card for communication between robots of the team and/or with a remote computer, and a Hokuyo URG-04LX laser range finder, which is the sensor used for implementing the Sensor-based Random Graph method. Laser scans are acquired at a 10 Hz rate and are characterized by an angular resolution of 0.36°, radial resolution of 0.1 cm, and maximum perception range R_p of 4 m. Since the scanning angle of the Hokuyo URG-04LX is 240°, when perceiving the robots perform a rotation on the spot to gather a 360° view.

B. Software and Control Architecture

Each Khepera III includes an Intel XSCALE PXA-255 400-MHz processor, with embedded Linux operating system, a 64-MB RAM, and a 32-MB Flash memory, that allow to implement onboard the SRG method (according to the software architecture of Fig. 4). In the debugging phase, however, only the robot driver runs on the robot, while the explorer process runs on a remote computer. During the exploration, an additional process, called *visualizer*, is in charge of "sniffing" and storing all the packets exchanged among the explorer processes in order to visualize and monitor the task progress.

A two-level architecture is adopted for controlling the motion of the robots. High-level velocity commands are issued at a 50 Hz rate by a trajectory tracking control scheme based on

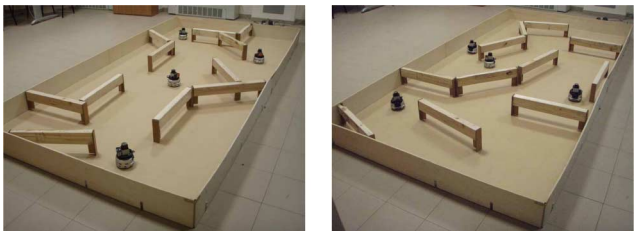

Fig. 17. (*Left*) Experiment 1. (*Right*) Experiment 2.

dynamic feedback linearization [27] and sent to the low-level PID controller of Khepera III, which is realized with a PIC18F4431. The reference trajectories are the SRG arcs generated by the action planner. In particular, in the following experiments, each arc is mapped to a 2-phase maneuver, in which the robot first rotates so as to point to the next node, and then travels to it along a line segment. A trapezoidal velocity profile is assigned over each phase.

Although a preliminary calibration of robot odometry and sensor parameters [28] was performed, dead reckoning proved to be inaccurate over relatively long paths, resulting in a degradation of the SRG method performance. Therefore, each robot has been provided with a basic self-localization module, in which incremental scan matching is used to correct odometric localization [29]. The information thus obtained is integrated in the high-level control law every five control cycles, due to the 10 Hz bound imposed by the scan acquisition frequency. In the presented experiments, the robots know their relative configurations at the start of exploration. Hence, mutual localization is maintained on the basis of this initial knowledge and self-localization data.

From a computational viewpoint, it is worth mentioning that the critical operations for the explorer process are graph managing and path search, while the most onerous operations for the robot driver are laser data acquisition and scan matching. Bandwidth is not an issue in our case, since the number of robots is limited. The used bandwidth is about 8 KB/s times the number of robots in the same subnetwork.

C. Results

The exploration environments were built inside a rectangular arena measuring 1.90×3.70 m. In view of the relatively small workspace, the perception range has been artificially limited to 1 m. The maximum cartesian velocity was set to 0.15 m/s. The robot communication range was unlimited (this is not required by the SRG method). In all the experiments, the workspace was completely covered and the robots terminated the exploration task by completing the homing phase.

Fig. 17 shows the environments used for the first two experiments, while Figs. 18 and 19 show the progress of the exploration and the SRG as reconstructed by the visualizer (a large red circle represents a robot, a small red point represents a node of the SRG) . The first environment is simply connected and its topology is correctly captured by the resulting SRG in the form of a tree (see Fig. 18). Instead, the multiply connected environment of the second experiment results in an SRG that is a proper graph (see Fig. 19). Note how the number of bridges (shown as dashed blue segments) in the second experiment is higher than in the first.

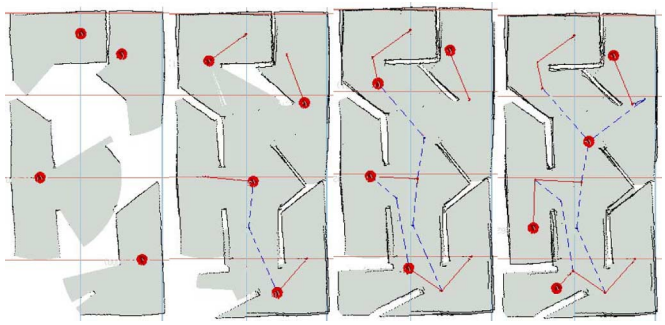

Fig. 18. Experiment 1 as reconstructed by the visualizer.

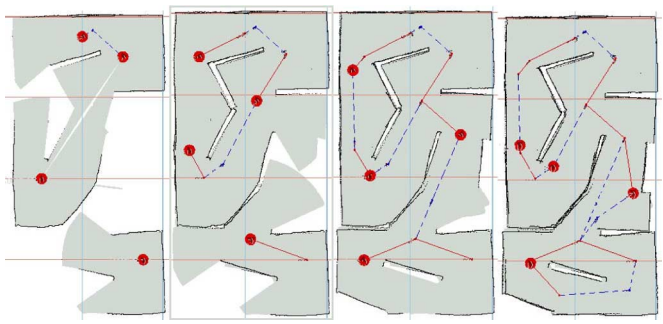

Fig. 19. Experiment 2 as reconstructed by the visualizer.

The overall performance of the SRG method in the two experiments is summarized in Table I. The first three rows of data quantify the contribution of each robot of the team to the exploration task, in terms of number of nodes, arcs, and bridges created by the robot. We also report the traveled distance, exploration time, and homing error for each robot. These data collectively indicate that a good degree of collaboration has been achieved by the team, as robots that added less nodes to the SRG traveled a shorter distance and terminated the exploration in less time (this means that there was no time wasted in visiting already explored regions). Also, the aggregate data statistics show that the exploration task has been distributed on the individual robots with a satisfactory degree of uniformity. The homing error is reasonably small and (obviously) tends to increase with the traveled distance.

The third experiment was carried out in the exploration environment shown in Fig. 20, using a team of only two robots starting from a clustered formation. Again, the topology of the environment was correctly captured by the resulting SRG that has the structure of a tree (see Fig. 21). Note how, during the exploration, one of the two robots moves along the main axis of the rectangular area, pushing the frontier of the explored region toward the boundary of the arena; the second robot trails along and completes the exploration by moving LFs that were created but not approached by the first robot.

TABLE I
TABLE OF NUMERICAL RESULTS FOR THE FIRST AND SECOND EXPERIMENT

	first experiment							second experiment						
	robots				aggregate data			robots				aggregate data		
	1	*2*	*3*	*4*	*mean*	*st dev*	*total*	*1*	*2*	*3*	*4*	*mean*	*st dev*	*total*
# nodes	5	7	2	3	4	2	17	4	5	7	3	5	2	19
# total arcs	5	7	1	5	5	3	18	4	5	9	3	5	3	21
# bridge arcs	6	4	0	0	2	2	7	2	2	6	2	3	2	12
traveled distance (m)	3.34	2.65	1.28	2.35	2.41	0.86	9.62	3.42	4.05	3.98	1.27	3.18	1.30	12.72
exploration time (sec)	240	228	92	174	184	67	240	265	259	264	179	242	42	265
homing error (m)	0.053	0.055	0.036	0.022	0.042	0.016	0.166	0.020	0.074	0.018	0.156	0.067	0.065	0.268

Fig. 20. Experiment 3.

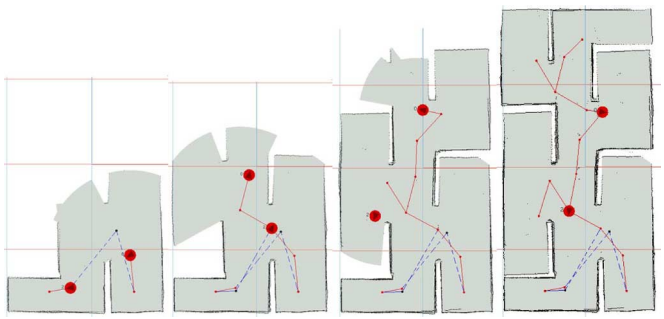

Fig. 21. Experiment 3 as reconstructed by the visualizer.

XI. CONCLUSION

In this paper, we have presented a decentralized strategy for cooperative robot exploration. A roadmap of the explored area, with the associated safe region, is built in the form of a compact data structure, called Sensor-based Random Graph. As it grows, the connectivity of the SRG is enhanced by adding bridges.

A simple and efficient decentralized cooperation mechanism is at the core of our method. This consists in an appropriate definition of the Local Frontier, by which each robot plans its motion toward areas that appear to be unexplored by the rest of the team on the basis of the available information. Local coordination guarantees that the collective motion of the team does not lead to collisions. Simulation and experimental results on a team of real robots have shown the satisfactory performance of the method both in ideal and practical conditions, even in the case of limited communication range.

We are currently working toward several objectives, among which we mention the following:

1) to develop a version of the SRG method for the case of nonomnidirectional sensors along the lines of [30], by extending the configuration vector so as to include the orientation of the robot;
2) to devise an SRG method for a team of heterogeneous robots, with different sensor capabilities and/or communication ranges;
3) to perform a quantitative study of the robustness and scalability properties of the method.

REFERENCES

[1] Y. Cao, A. Fukunaga, and A. Kahng, "Cooperative mobile robotics: Antecedents and directions," *Auton. Robots*, vol. 4, no. 1, pp. 7–27, 1997.

[2] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes, "A taxonomy for multi-agent robotics," *Auton. Robots*, vol. 3, pp. 375–397, 1996.

[3] I. Rekletis, G. Dudek, and E. Milios, "Multi-robot collaboration for robust exploration," *Ann. Math. Artif. Intell.*, vol. 31, no. 1, pp. 7–40, 2001.

[4] D. Goldberg and M. Mataric, "Interference as a tool for designing and evaluating multi-robot controllers," in *Proc. 14th AAAI/9th IAAI*, 1997, pp. 637–642.

[5] T. Balch and R. Arkin, "Communication in reactive multiagent robotic systems," *Auton. Robots*, vol. 1, no. 1, pp. 27–52, 1994.

[6] B. Yamauchi, "Frontier-based exploration using multiple robots," in *Proc. 2nd Int. Conf. Auton. Agents*, 1998, pp. 47–53.

[7] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, "Coordinated multi-robot exploration," *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 376–386, Jun. 2005.

[8] A. Howard, L. Parker, and G. Sukhatme, "Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection," *Int. J. Robot. Res.*, vol. 25, no. 5/6, pp. 431–447, 2006.

[9] J. Ko, B. Stewart, D. Fox, K. Konolige, and B. Limketkai, "A practical, decision-theoretic approach to multi-robot mapping and exploration," in *Proc. 2003 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 3232–3238.

[10] A. Howard, M. Mataric, and S. Sukhatme, "An incremental deployment algorithm for mobile robot teams," in *Proc. 2003 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 2849–2854.

[11] R. Zlot, A. Stenz, M. Dias, and S. Thayer, "Multi-robot exploration controlled by a market economy," in *Proc. 2002 IEEE Int. Conf. Robot. Autom.*, pp. 3016–3023.

[12] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. Younes, "Coordination for multi-robot exploration and mapping," in *Proc. 17th AAAI/12th IAAI*, 2000, pp. 852–858.

[13] A. Franchi, L. Freda, G. Oriolo, and M. Vendittelli, "A randomized strategy for cooperative robot exploration," in *Proc. 2007 IEEE Int. Conf. Robot. Autom.*, pp. 768–774.

[14] A. Franchi, L. Freda, G. Oriolo, and M. Vendittelli, "A decentralized strategy for cooperative robot exploration," in *Proc. 1st Int. Conf. Robot Commun. Coordination*, 2007.

[15] G. Oriolo, M. Vendittelli, L. Freda, and L. Troso, "The SRT method: Randomized strategies for exploration," in *Proc. 2004 IEEE Int. Conf. Robot. Autom.*, pp. 4688–4694.

[16] L. Freda and G. Oriolo, "Frontier-based probabilistic strategies for sensor-based exploration," in *Proc. 2005 IEEE Int. Conf. Robot. Autom.*, pp. 3892–3898.

[17] J.-C. Latombe, *Robot Motion Planning*. Norwell, MA: Kluwer, 1991.

[18] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi, *Reasoning about Knowledge*. Cambridge, MA: MIT Press, 1995.

[19] J. Serra, *Image Analysis and Mathematical Morphology*. New York: Academic, 1982.

[20] H. H. Gonzalez-Banos and J.-C. Latombe, "Navigation strategies for exploring indoor environments," *Int. J. Robot. Res.*, vol. 21, no. 10, pp. 829–848, 2002.

[21] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.
[22] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: Freeman, 1983.
[23] F. Harary, *Graph Theory*. Reading, MA: Addison-Wesley, 1994.
[24] A. Meijster, J. Roerdink, and W. Hesselink, *Mathematical Morphology and Its Applications to Image and Signal Processing*. Norwell, MA: Kluwer, 2000.
[25] A. Franchi, G. Oriolo, and P. Stegagno, "Mutual localization of a multi-robot team with anonymous relative position measures," Dept. Comput. Syst. Sci. "A. Ruberti," Tech. Rep. 1/2009, 2009.
[26] T. Simeon, J.-P. Laumond, and F. Lamiraux, "Move3d: A generic platform for path planning," in *Proc. 4th Int. Symp. Assem. Task Planning*, 2001, pp. 25–30.
[27] G. Oriolo, A. D. Luca, and M. Vendittelli, "WMR control via dynamic feedback linearization: Design, implementation, and experimental validation," *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 6, pp. 835–852, Nov. 2002.
[28] A. Censi, L. Marchionni, and G. Oriolo, "Simultaneous maximum-likelihood calibration of robot and sensor parameters," in *Proc. 2008 IEEE Int. Conf. Robot. Autom.*, pp. 2098–2103.
[29] A. Censi, "An ICP variant using a point-to-line metric," in *Proc. 2008 IEEE Int. Conf. Robot. Autom.*, pp. 19–25.
[30] L. Freda, G. Oriolo, and F. Vecchioli, "Sensor-based exploration for general robotic systems," in *Proc. 2008 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 2157–2164.

Antonio Franchi (S'07) received the M.Sc. degree in electrical engineering in 2005 from the Università di Roma "La Sapienza," Rome, Italy, where he is currently working toward the Ph.D. degree.

His current research interests include the areas of multirobot control and state estimates, with applications to exploration, localization, and motion coordination problems.

Luigi Freda (S'05–M'07) received the M.Sc. degree in computer engineering and the Ph.D. degree in control systems engineering from the Università di Roma "La Sapienza," Rome, Italy, in 2003 and 2007, respectively.

In 2006, he was a Visiting Scholar for six months at the Motion Strategy Laboratory, University of Illinois at Urbana-Champaign (UIUC), Urbana. He is currently a Research Associate in the Department of Computer and Systems Science (DIS), Università di Roma "La Sapienza." His current research interests include the areas of robot exploration, motion planning, minimal sensing, and control of mobile robots.

Giuseppe Oriolo (S'89–M'92–SM'02) received the Ph.D. degree in control systems engineering from the Università di Roma "La Sapienza," Rome, Italy, in 1992.

He is currently an Associate Professor of automatic control and robotics at the Università di Roma "La Sapienza," where he also heads the Robotics Laboratory. His current research interests include nonlinear control and robotics, visual servoing, redundant manipulators, mobile and nonholonomic robots, motion planning, sensor-based navigation and exploration, and service robotics.

Dr. Oriolo is an Editor of the IEEE TRANSACTIONS ON ROBOTICS.

Marilena Vendittelli (M'09) received the M.Sc. degree in electrical engineering and the Ph.D. degree in control systems engineering from the Università di Roma "La Sapienza," Rome, Italy, in 1992 and 1997, respectively.

From April 1997 to October 1998, she was a Postdoctoral Fellow at the Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS), Centre National de la Recherche Scientifique (CNRS), Toulouse, France. She is currently an Assistant Professor in the Department of Computer and Systems Science (DIS), Università di Roma "La Sapienza." Her current research interests include the area of robot motion planning and control, with particular emphasis on nonholonomic and redundant systems, navigation and perception for mobile robots, and multirobot exploration.