The Robot Swarm Re-localization Problem

Nicolas Bredeche, Yann Chevaleyre

HAL Id: inria-00336690
https://inria.hal.science/inria-00336690

Submitted on 4 Nov 2008

The Robot Swarm Re-localization Problem

Nicolas Bredeche
TAO / LRI ; Univ Paris-Sud, CNRS, INRIA Futurs
Université de Paris-Sud - LRI, Bat. 490 - Orsay, France
bredeche@lri.fr

Yann Chevaleyre
LAMSADE
Université Paris Dauphine - Paris, France
yann.chevaleyre@lamsade.dauphine.fr

Abstract—**This paper tackles the problem of re-localization in a swarm of moving virtual robots where some robots might be lost and where the distance between each neighboring robot is known within a limited communication radius (e.g. using the strength of a radio/wifi signal), whereas the precise localization is not known. The basic process in relocalization relies in a robot swarm that re-aggregates based on local information only by recruiting lost robots so as to build a swarm configuration where localization is possible and non-ambiguous. The main issues in relocalization is to provide highly decentralised behaviors for ensuring efficient and fast re-aggregation of the swarm that is scale independent. In this paper, the problem of relocalization is defined as well as criteria for evaluating swarm relocalization efficiency. Moreover, a set of decentralized behaviors based on local reactive behaviors is presented and experimentally studied.**

I. Introduction

The advantages of relying on a swarm of robots rather than on a single robot are clear for a wide range of applications, would it be because of the intrinsic reliability of distributed agents or because of the parrallel approach to solving problems. However, considering a swarm of robots also implies to deal with a network of robots that should be synchronised in a distributed fashion so as to achieve organism-level tasks.

A key feature of swarm formation is to rely on coordination and cooperation on the local scale, possibly considering limited communication and computation ability, with regards to an objective (e.g. pattern formation) specified on the global scale. Such approches have long been advocated since early works in this domain[5][4] for a wide range of applications ranging from flocking to patrolling and cooperation. While these early works were application-driven (flocking, coordination, transport, etc.), more recent works have been addressing higher level approaches that can provide a basis for targeted applications as well as useful information for higher level components such as Mapping algorithms. The originality of these recent works is to consider properties regarding swarm or multi-robot configuration and organization strategy from a graph theory perspectives.

In the scope of this paper, the problem of re-localization in a swarm of robots is addressed. Given a swarm of robots, the goal is to achieve a configuration of connected robots that makes possible non-ambiguous localization of each robot in the swarm, given the robot have limited computation and communication capabilities. This implies both building a decentralized coordinate system and localizing each of the robot with regards to a base reference in the context of robot using radio signal that can be emitted/received within a limited range. As proposed in recent works[2], swarm localization is reformulated as the construction of a specific sub-class of grounded graphs that ensures localizability (ie. a swarm configuration from which actual localization can be handled by a dedicated algorithm, e.g. [1]). The central contributions of this paper are (1) to define the problem of re-localization of a swarm of moving robots, ie. strategy for recruiting new "lost" robots into the swarm for further localization, (2) to define some relevant criteria for evaluating re-localization and (3) to provide a set of behaviors for ensuring re-localization based on local information that can be applied in a distributed fashion. As the problem of re-localization implies moving robots, issues such as swarm coverage of the environment as well as behavior issues for "lost" robots are carefully studied.

The next section gives some definition of the problem of localization as well as its formulation as a problem of localizability in grounded graph. Section 3 describes the approach towards re-localization, which is based on incrementally re-building a swarm configuration that is shown to belong to a restricted family of localizable grounded graph. This approach is experimentally studied as well as analysed and discussed in section 4. Finally the last section concludes and gives some perspectives.

II. Problem Setting and Related Works

In the scope of robot swarms, the amount of reliable information available to each robot may be relatively limited. A typical example is that of using wifi or radio signals so as to localize neighboring robots, which makes it possible to perceive information within a given radius, with a monotonously decreasing signal power. This implies that information contained by the radio signal include both data exchange as well as distance information. However, radio signals have radial diffusion which prevent a robot to stricly localize its neighbors. As a consequence, localization with regards to neighbors must be the results of a triangulation process using incoming signals from at least three robots (in 2D).
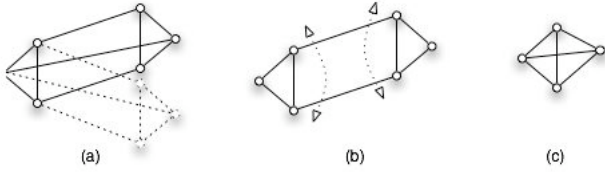
Fig. 1. Three grounded graphs. (a) and (b) are not localizable, whereas (c) is.

Localization through triangulation, or "trilateration", is a rather classic way of proceeding in the field of mobile sensor networks, would it be for localizing mobile phones in telecom networks or one-to-one mobile robots localization for environment coverage[3]. However, specificities of robot swarms imply the distributed flavor of localization as well as a limited range of radio signals compared to the swarm size. This implies that localization must be possible accross the swarm thanks to some self-contained *beacon*, which refers to a limited number of robots (usually, 3 robots) whose relative positions and inter-connection makes it possible to act as a local base for localizing neighboring robots, and some kind of *relays* that enables localization of distant robots.

From the graph theory perspective, a swarm of robot with radio connections giving information about distances can be seen as a grounded graph, defined as follows:

Definition 1: A *grounded graph* [6] is a graph $G(V, E)$ with edges labelled with d_{ij}, such that each vertex represents a robot, and two vertices (i, j) in the graph are connected if the distance d_{ij} between them is known.

Thus, given a set of robots connected together with radio links, such that two connected robots have the ability to estimate the distance between them, then a corresponding grounded graph can be build with robots as vertices and weighted edges as signal links with distance. Localizing the agents consists in computing a *realization* of this grounded graph:

Definition 2: A *realization* [7] of a grounded graph G is a function that maps vertices of G to points in the euclidian space, such that for each edge (i, j) of G, the euclidian distance between the two corresponding points is equal to d_{ij}. A grounded graph is *realizable* if some realization exists. Moreover, G is said to be *localizable* iff all possible realizations are equal up to some translation, rotation, and reflection of the entire space.

Figure 1 illustrates grounded graphs that are, or not, localizable. The major problems are that of continuous deformations (case (b), the graph may be continuously deformed) and rigid deformations (case (a), the graph may flip from one realization to another). On the other hand, case (c) gives an example a localizable graph, which is formally defined as 3-connected and redundantly rigid[7]. A direct consequence of dealing with such localizable graphs is that it is possible

to compute coordinates for each robot with regards to one another.

Although checking the localizability is known to be NP-hard in general, under a few natural assumptions, a polynomial algorithm is known for this problem[7]. It can be observed that checking for localizability in the general case implies computation on the global scale since all the links between robots have to be considered to ensure complete check. Because of the intrinsic nature of swarms, this is of course a problem since one has to consider local computation and peer-to-peer communication, thus enforcing specific localizability and localization algorithms.

Even harder than checking localizability, the problem of computing the realization has been shown to be NP-hard *even with strong restrictions on grounded graphs* [6] and even in one-dimension[7]. As a consequence, many works have focused on defining simple sufficient (but not necessary) conditions under which localizability is ensured and computing this realization becomes tractable. Among such work, Eren et al. define *trilateration*[6] :

Definition 3: A *trilaterative ordering* [6] of a grounded graph $G(V, E)$ is an ordering of the vertices $v_1, \ldots, v_n$ such that the first three vertices are connected together, and that for all $i \in 4 \ldots n$, v_i is connected to at least 3 vertices appearing before v_i in the ordering. Grounded graphs for which a trilaterative ordering exist are called *trilateration graphs*.

It was shown in [6] that trilateration graphs are localizable. Thus, once a trilaterative ordering has been found on some grounded graph, the two above problems are easily solved.

Note that in trilaterative orderings, the first four nodes form a quadrilateral (a set of four fully connected vertices). The key feature of such a "quad" is that it is the *only* pattern of 4 vertices that ensure localizability (cf. definition 2). This means that given a set of four interconnected reference robots - a quad - it is possible to localize all robots in a non-ambiguous fashion with regards to this reference robots (ie. an absolute coordinate system). Similarly, the approach presented in [2] addressed this issue by considering specific swarms of robots build through quadrilateration of each node. Their approach consist of two steps: firstly quads are grown locally on the graph and connected to each other only if two quads share three vertices (to ensure localizability), secondly, quad clusters are connected with one another through shared vertices, which may imply global message exchange and costly computation to trilaterate clusters among themselves (e.g. two clusters sharing three different vertices with each vertex being part of different quads involves long distance communication to solve the localization problem). Clearly, this two steps approach can be seen as building a trilaterative ordering. Thus, when this whole process succeeds, the graph is ensured to be localizable and the localization to be easily computable.
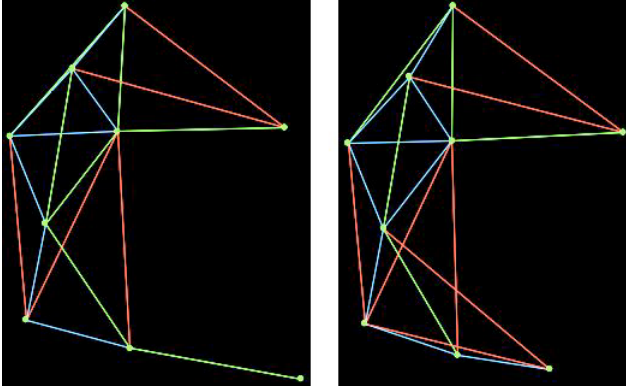
Fig. 2. A swarm before and after re-localization. (left) Swarm partially localized. The agent at the bottom-right corner has a single neighbor, which is not sufficient for him to compute its position by triangulation. (right) After re-localization process, the swarm has rearranged itself: agent at the bottom-right has moved closer to the cluster, thereby connecting itself to three agents. Clearly, the swarm is now trilateralizable.

As in most works focusing on the localization problem, the approach of [2] relies on the choice of a specific growing pattern (quads) growing on the grounded graph. This impacts the family of localizable grounded graphs one has to consider. As a consequence, it may exclude candidate graphs, or at least may imply costly computation to overcome exceptions which might be more or less frequent in the real world. Thus, in many cases in which the graph is localizable, this approach will not succeed in finding the realization.

This also has some impact regarding the problem of re-localization. Indeed, the problem of re-localization considers a swarm in a localization mode with moving robots and can be formalized as follows:

Definition 4: Let G be a grounded graph associated with set of robots. Given that G is not localizable, the re-localization problem consists in rearranging the swarm of robots in such a way that localizability is restored.

In this context, the key issues are to consider both localizability within clusters of inter-connected agents as well as a strategy regarding recruitment of robots having too few connections to be localizable. Figure 2 shows an example of swarm being re-localized.

Four major steps constitute re-localization: (1) choosing a basic node which will serve as a refence (possibly through automatic bidding process) ; (2) growing a non-ambiguous pattern as a basis for trilateration (3) extending this pattern to the neighborhood when trilateration is possible (4) recruiting new (possibly moving) robots that are near-by but cannot be directly trilaterated.

Re-localization is thus largely dependant on the localization strategy (constraints over the grounded graphs considered), which also implies the relation between swarm configuration and recruiting lost robots (ie. the wider the swarm coverage, the more chance to recruit new robot). In

the following sections, some criteria are described to evaluate the efficiency of re-localization in the scope of considering swarm as grounded graph with trilaterative ordering as well as several behaviors that can be distributed among the swarm.

III. LOCALIZABILITY AND RE-LOCALIZATION

A. Criteria for Evaluating Relocalization

In order to evaluate to performance of relocalization algorithms, a set of evaluation criteria has to be devised. These criteria may encompass several, possibly conflicting, objectives which are to be selected depending on the task at hand. Firstly, the following terms are defined: let (x_i, y_i) be the position of agent i at the beginning of the relocalization process, and (x'_i, y'_i) be its position when relocalization is achieved. Next, a set of generic criteria is proposed:

- **C0**. Relocalization Speed: time elapsed until localizability is ensured.
- **C1**. Agents Displacement: sum of the distances between each agent's position before and after relocalization, namely $\sum_i distance\left((x_i, y_i), (x'_i, y'_i)\right)$
- **C2**. Agent with Maximal Displacement: max euclidian distance between each agents position before and after relocalization, namely $\max_i distance\left((x_i, y_i), (x'_i, y'_i)\right)$
- **C3**. Agents Involvement: number of agents which will have to move to ensure localizability.
- **C4**. Communication load: number of bits exchanged between agents during relocalization.
- **C5**. Neighborhood co-linearity. Average value during relocalization of the following algorithm: for each agent, estimate to what extent connected neighbors are +/- co-linears. This can be an important issue to localization as linear independence is required for ensuring localization.

Underlined items are considered in the next experimental section - indeed, C0 and C1 are the most relevant straight-forward criteria to optimize when it comes to relocalization. However, other criteria may be of interest depending on the context. As examples, C3 might be the most important one in the scope of rescue robots while C4 is particularly interesting in the context of low energy consumption (note that in this paper, it is not relevant as we consider behavior based on local information).

B. Generic Algorithms for Relocalization

The re-localization problem implies to consider three classes of agents depending on their respective situation regarding localizability, and are defined as follow:

- **LOC** - Localized agents: A cluster of agents for which a trilaterative ordering has been computed. As a consequence, LOC agents can be localized relatively to each other. Clearly, $count(LOC_{neighbors}) \geq 3$, with $LOC_{neighbors}$ the set of all LOC agents connected to the current agent).

- **CNL** - Connected but Not Localized agents: agents that are connected to at least one LOC agent, and at most two LOC agents. ie. their aren't enough edges towards the cluster of LOC agents to allow CNL agents to compute their relative position (ie. $count(LOC_{neighbors}) \in \{1, 2\}$).
- **NC** - Not Connected agents: NC agents may have neighbors, but none of them are LOC agents. Ie. NC agents have strictly no information concerning their position (ie. $count(LOC_{neighbors}) = 0$).

Broadly, this means that LOC agents are part of a "localizable cluster" while CNL and NC agents are considered as two kinds of "lost agents". These classes of agents also differs by the relocalization behavior used so as to reach global swarm localizability. As a consequence a class of behaviors is defined for each class of agents in order to described the type of behaviors used to address re-localization. Namely, the following classes of behaviors are defined: LOC-behaviors, CNL-behaviors and NC-behaviors. Thus, the basic algorithm for relocalization, "relocalization algorithm", is distributed on each agent and is summarized as follow:

- If LOC-agent Then
-    call-LOC-behavior
- Else
-    If CNL-agent Then
-      call-CNL-behavior
-    Else
-      call-NC-behavior
-    End If
- End If

At the beginning of the re-localization process, as no trilaterative ordering has been computed, the LOC class does not contain any agent. Thus, it now remains to describe how to bootstrap the relocalization by selecting the first three LOC agents from which the whole localizable cluster can be incrementally built. As shown in the previous section, constructing a trilaterative ordering on a grounded graph suffices to solve the localization problem, when this construction is possible. The "incremental trilateration bootstrap algorithm" is described as follow: Let LOC-agent$_{source}$ be a robot defined as the unique starting point to grow the localizable swarm (see below). For any robot in the swarm that is both directly connected to LOC-agent$_{source}$ and not yet a LOC-agent itself (ie. this is a CNL-agent), check with each neighboring robots (with the exception of LOC-agent$_{source}$) if it is also connected with LOC-agent$_{source}$ - if checking succeeds, then agent state switches to LOC-agent. To sum it up, this means that each possible candidates for boostraping localization simply check if they can take part in a group of three inter-connected robots, one of them being the LOC-agent$_{source}$. Afterwards, the algorithm described previously makes it possible to relocalize the whole swarm through incrementally building a trilaterative ordering.

The selection of a specific robot as LOC-agent$_{source}$ can be performed through various process, some of which are given as examples here: ad hoc selection of a reference robot, bidding process between all robots prior to deployment, human-triggered when relocalization is needed (e.g. a human supervisor following the robots equipped with a short range emitter), etc. The last example is of particular interest as human supervision in the context of a swarm of robots also face the problem of highly distributed agents over a terrain and may need to trigger flocking and relocalization behavior to centralize and gather important data - this is a typical example of exploration of indoor environment after natural disasters where such a property of relocalization from a robot selected on the fly is needed.

Note that ensuring localizability is very different from computing localization: maintaining localizable swarm implies only local maintenance of connectivity between robots (ie. avoid breaking links with the neighborhood) while localization itself implies synchronization and message exchange of all individuals in the swarm as the coordinate system is highly dependant of swarm relative position and movement. In a real world experiment, localization, which implies costly communication, may be performed from time to time when necessary while localizability, which implies very few (and only local) messaging, must be ensured any time.

C. Agent Behaviors for Relocalization

As stated before, the goal of NC agents is to find CNL/LOC agents so as to connect to the localizable group of robots. Having no information whatsoever, they must explore their environment. When encountering other NC agents, they might exchange information about where to go/not to go. In the following, several candidate behaviors are described for each the agent class.

1) NC behaviors:
- **RE**. Random forward exploration. The agent moves forward and rotates around itself with small random angle.
- **BOIDS**. Agent follows the boids algorithm as described in [9], and tend to aggregate while exploring the environment.

2) CNL behaviors:
- **RE**. Random forward exploration (same as before).
- **CLOSER-MAXLOC**. The agent gets closer to the LOC agent of maximal degree.

3) LOC behaviors:
- **STATIC**. Agent does not move.
- **COVER**. The agent gets away from closest LOC agent if closer than a given pre-defined *threshold_distance*. With this behavior, LOC agents tend to spread out.

Each of these behaviors are reactive behaviors that rely solely on local information, implying minimal (ie. zero) communication accross the swarm.
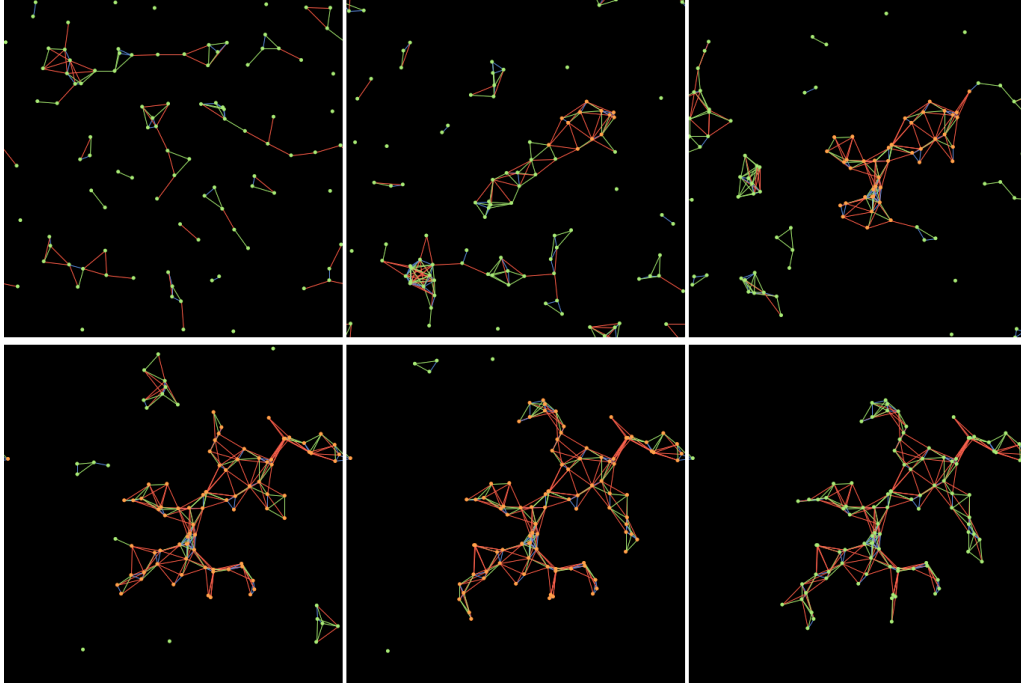
Fig. 3. Swarm relocalization through time (from upper-left to bottom-right) with RE behaviors for NC and CNL robots and STATIC behavior for LOC robots. Both robots and *comunication links* between robots are shown in this figure. The resulting configuration holds the desired unique realization property.

NC/CNL/LOC	radius=6		radius=9		radius=12	
behaviors	C0	C1	C0	C1	C0	C1
RE,RE,STATIC	777 (18)	355 (20)	308 (9)	108 (3)	96 (2)	39 (2)
RE,RE,COVER	*846 (19)*	*415 (29)*	*387 (11)*	*135 (4)*	*145 (6)*	*53 (3)*
RE,CLOSER-MAXLOC,STATIC	589 (18)	259 (15)	211 (6)	85 (4)	58 (2)	23 (1)
RE,CLOSER-MAXLOC,COVER	643 (22)	*306 (21)*	228 (7)	85 (4)	60 (2)	24 (1)
BOIDS,RE,STATIC	651 (17)	215 (17)	247 (8)	79 (2)	81 (4)	34 (2)
BOIDS,RE,COVER	669 (16)	221 (6)	322 (13)	102 (5)	125 (7)	43 (3)
BOIDS,CLOSER-MAXLOC,STATIC	**515 (16)**	**176 (5)**	**160 (5)**	**58 (2)**	**47 (2)**	**19 (1)**
BOIDS,CLOSER-MAXLOC,COVER	566 (19)	196 (10)	163 (6)	61 (2)	48 (2)	21 (1)

TABLE I
EXPERIMENTS WITH DIFFERENT BEHAVIORS FOR NC, CNL AND LOC ROBOTS USING A PERCEPTION RADIUS OF EITHER 6, 9 OR 12. EVALUATION OF RELOCALIZATION USING C0 AND C1 CRITERIA. FOR EACH EXPERIMENTS, AVERAGE VALUE OVER 80 RUNS AND STANDARD DEVIATION IS PROVIDED. BEST RESULTS FOR EACH RADIUS SIZE AND CRITERIA IS IN BOLD FONT. WORST RESULTS ARE MARKED WITH ITALIC FONT.

IV. EXPERIMENTS AND DISCUSSION

In this section, a set of experiments implying several combinations of aforementionned behaviors were tested in different setups. All experiments were run using the NetLogo framework, which has long been considered as a major development platforms for simulating robot swarms[10]. The environment is defined as a (continuous) square area of size 100×100 with wrapping enabled (ie. no border). In all experiments, 100 agents were randomly placed on the area. Each agent is connected to all agents within a radius of either 6, 9 or 12. In order to evaluate the average performance for all settings, all experiments are performed over 80 runs so as to provide smooth results. As stated in the previous section, several combination of behaviors are put to the test

so as to provide 8 differents behavior combinations (LOC: Static or Cover, CNL: RE or CLOSER-MAXLOC, NC: RE or BOIDS). Given the three possible radii, this means that 24 distinct experimental setups are defined. Figure 3 shows a typical swarm relocalization process through time, ie. from the initial swarm configuration with many lost robots until a fully localizable swarm.

Results for all experiments are presented in table I using two generic criteria: C0 (time taken to complete relocalization) and C1 (sum of agent displacements). Random exploration behaviors, both for CNL and NC robots, produce the worst results, whatever the LOC robot behavior used. This is not really a surprise as these setups are considered as the baseline approach due to its simplicity. Then, it

can observed that both CNL "CLOSER-MAXLOC" and NC "BOIDS" behaviors separately improve results on C0 as the perception radius of robots grows. Indeed, using CNL "CLOSER-MAXLOC" tends to lead to even better results as the radius grows, which may vote in favor of the crucial importance of an efficient behavior on the edge of relocalizability. Then, combining both behaviors leads to the best results observed, whatever the LOC behavior considered. Interestingly enough, C0 and C1 are strongly correlated, with the notable exception when using a small radius and the CNL "CLOSER-MAXLOC" behavior (C1 is worse than expected compared to other experiments and the current C0 value). On the other hand, the LOC "COVER" behavior have a more or less important negative impact, which is mainly due to deadlock situations because of limitations of coverage algorithms where each LOC agents must remain connected to all its LOC neighbors (ie. agents get stuck in a situation were they cannot move anymore).

The take home message from these experiments is that relocalization seems to depend first on accurate CNL behaviors, for which there exists some local information that can be efficiently exploited, and then on NC behaviors, for which few information is available.

V. Conclusions and Perspectives

In this paper, we introduced the problem of *re-localization*, which extends the localization task to moving, possibly lost, agents. The motivation behind relocalization is to provide an efficient algorithm to ensure swarm localizability, ie. the fact that a localization algorithm provide a unique non ambiguous position for each of the robot in the swarm with regards to the whole swarm. The major contributions of this paper are to provide a formal definition of the relocalization problem as well as proposing a set of evaluation criteria and a methodology towards defining a relocalization strategy. Moreover, a set of local behaviors addressing relocalization are presented. These behaviors are designed so as to be distributed over the robots in a swam. An experimental study as well as an in-depth analysis is provided. The major advantage of these behaviors is to require very low bandwidth by using only local information.

Relocalization is a key feature in swarm robotics, as it addresses the problem of dynamically recovering a swarm configuration where localization algorithms may be used. Moreover, many open questions are raised in this paper, some of which are listed below:

- Currently, behaviors based on local interaction have been used. However, there might be a trade-off between bandwidth consumption and relocalization efficiency. For example, the class of gossip behaviors described previously may enable lost robots to avoid already explored areas.

- Noisy measures regarding distance to neighbors impact the spatial distribution of robots. This is also a problem in the scope of the localization problem where connected, but aligned, robots fail to provide a reliable basis for localizing other robots. This problem has already been addressed to some extent for localization, for example in [2], and could provide a starting point for designing noisy re-localization algorithms. Moreover, the reality gap between experiments in simulation and actual implementation on real robots should definitely be addressed in further works.

- Re-localization may be stated as an optimization problem - which imply designing the optimization criteria to be used. Indeed, criteria defined in this paper provide a starting point, yet far from complete.

- More formally, efficient centralized algorithm for solving re-localization should be studied so as to provide a basis comparaison for distributed algorithms. In a similar fashion, the computational complexity of re-localization should be carefully studied.

References

[1] Thomas F., Ros L. Revisiting trilateration for robot localization. IEEE Transactions on Robotics, Volume 21, Issue 1, p.93-101, 2005.

[2] Moore D., Leonard J., Rus D. and Teller S. Robust Distributed Network Localization with Noisy Range Measurements. In Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys '04), pp. 5061, Baltimore, MD, November 2004.

[3] Maria Gini and Richard Voyles. Robotic Swarm Dispersion Using Wireless Intensity Signals. Distributed Autonomous Robotic Systems 7. 2006.

[4] Dudek G., M. Jenkin, Milios E. and Wilkes D. A taxonomy for multi-agent robotics. Autonomous Robots, Vol. 3, 375-397, 1996.

[5] Bayindir L. and Sahin E. A review of studies in swarm robotics. Turkish Journal of Electrical Engineering, Vol.15, no.2, 2007.

[6] T. Eren, D. Goldenberg, W. Whitley, Y. Yang, A. Morse, B. Anderson, and P. Belheumer. Rigidity, Computation, and Randomization of Network Localization. In Proc. IEEE Conf. Computer Comm. (Infocom '04), Mar. 2004.

[7] B. Hendrickson. Conditions for unique graph realizations. SIAM J. Computing, 21(1992), pp. 65–84. 1992.

[8] Molloy, M., Reed, B. The Size of the Giant Component of a Random Graph with a Given Degree Sequence. Combin. Probab. Comput. 7(1998), 295-306.

[9] Reynolds, C.W. Flocks, Herds and Schools: A Distributed Behavioural Model. ACM Computer Graphics, 21, (SIGGRAPH 1987 Conference Proceedings), 25-34. 1987.

[10] Railsback, S. F., Lytinen, S. L., and Jackson, S. K. 2006. Agent-based Simulation Platforms: Review and Development Recommendations. Simulation 82, 9 (Sep. 2006), 609-623. 2006.