# Collaborative Coverage using a Swarm of Networked Miniature Robots

Samuel Rutishauser [a] Nikolaus Correll [b] Alcherio Martinoli [a]

[a] *Distributed Intelligent Systems and Algorithms Laboratory*
*École Polytechnique Fédérale Lausanne, CH-1015 Lausanne, Switzerland*

[b] *Computer Science and Artificial Intelligence Laboratory*
*Massachusetts Institute of Technology, 02139 Cambridge, MA, USA*

**Abstract**

We study distributed coverage of environments with unknown extension using a team of networked miniature robots analytically and experimentally. Algorithms are analyzed by incrementally raising the abstraction level starting from physical robots, to realistic and discrete-event system (DES) simulation. The realistic simulation is calibrated using sensor and actuator noise characteristics of the real platform and serves for calibration of the DES microscopic model. The proposed algorithm is robust to positional noise and communication loss, and its performance gracefully degrades for communication and localization failures to a lower bound, which is given by the performance of a non-coordinated, randomized solution. Results are validated by real robot experiments with miniature robots with a size smaller than 2cm x 2cm x 3cm in a boundary coverage case study. Trade-offs between the abilities of the individual platform, required communication, and algorithmic performance are discussed.

*Key words:* Distributed Coverage, Networked Robotic Systems, Modeling of Multi-Robot Systems

## 1 Introduction

Multi-robot coverage [1–7] requires the coordination of a team of robots to entirely cover an environment with a specific end-effector (sensor or actuator) mounted on each robot. Different metrics are possible, the most common being

⋆ This work has been done when all authors were with the Swarm-Intelligent Systems Group, EPFL

coverage duration and coverage redundancy. Applications are cleaning [8], demining [9], or inspection of structures [10], to name a few. In the case of previously unknown environments, complete coverage goes side by side with complete exploration of the environment [3,5,7,11]. In this case, an (optimal) coverage path cannot be planned in advance.

When compared with a single robot approach, a multi-robot solution potentially offers advantages in terms of robustness toward failures, expedited task completion due to working in parallel [1, 3, 4, 7], or improved accuracy by exchanging sensory information [7]. Complete coverage of an area by a tool mounted on the robot should not be confused with mapping an environment which requires complete coverage of the environment by a long-range sensor [11, 12], or the permanent coverage of an area by a set of sensors [13] for surveillance and environmental measurement tasks.

### *1.1 Contribution of this paper*

We present an algorithm that leads to complete coverage of a previously unknown environment using miniature robots subject to severe sensor and actuator noise. The algorithm is fully distributed and requires only local communication. The algorithm is robust toward localization errors and communication failure and allows us to calculate lower bounds for the probability to complete. In particular, the performance degrades gracefully under influence of sensor and actuator noise and is demonstrated on miniature robots with extremely limited localization, navigation, and communication abilities.

We study the effect of erroneous localization information and limited range communication on the coverage performance experimentally and by modeling the system at multiple levels of abstractions. The models are calibrated by measuring low-level parameters such as sensor and actuator noise on the real hardware and higher level parameters, such as the probability of successfully completing a certain behavior, in realistic simulation. By explicitly modeling sensor and actuator noise, we show that the dynamics of a complex distributed system can be quantitatively correct predicted.

### *1.2 Related work*

Algorithmic performance and controller design for a multi-robot system are a function of the capabilities of the individual platform (e.g., in terms of perception, actuation, communication) and the system as a whole (e.g., amount of a priori knowledge, centralized vs. decentralized approaches), which makes comparison of different algorithms using a single metric difficult.

Regardless of the complexity of the chosen approach for coordination in terms of computational and communication requirements, the ability to perform a more or less crude on-line cellular decomposition of the environment is the basis for most coverage algorithms (see [14] for an overview) except those that are fully probabilistic (see [10] for a comparison on a suite of fully reactive algorithms) or those where the environment is completely known in advance [3, 4], which allows for decomposing the environments into cells off-line. For online cellular decomposition, some algorithms only require bumper sensors [15], assuming rectinlinear environments, while others use long range sensors [7, 9, 11, 12].

An important question is also, whether robots will make up for potential failures of themselves or of other robots as in e.g. [2, 12] or [16]. Sources of error include both sensor and actuator noise. Sensor noise is usually addressed by associating probabilities for coverage or exploration with parts of the environment (e.g., using occupancy grids as in [12]), whereas actuator noise and complete robot failure are usually faced by deriving provably complete policies for coordination (as in [2, 16]) that require at least one robot not to fail. Notice that coping with sensor and actuator noise are dual problems, i.e. it is difficult to detect one when the other is also uncertain. For this reason both [2, 16] assume perfect localization. An attempt to solve this problem is to use simultaneous localization and mapping (SLAM) techniques [12, 17]. Here, recent sensor information is matched to the stored occupancy grid in order to achieve a maximum likelihood estimate of the robot's position. This approach, however, requires the availability of vast sensory information and computational power and does not completely resolve uncertainty.

With respect to coordination, we distinguish approaches that collaborate simply by exchanging information [12] or explicitly arbitrate coverage tasks among robots [4,16], which usually require extensive communication among the agents. In [16], an auction-based algorithm [18] is used for arbitrating coverage among the robots. Calculating bids, which involves solving an instance of the traveling salesman problem for an optimal solution [10], however, requires additional computational capabilities of miniature robotic platforms such as those considered in this paper.

## 2 Networked Multi-Robot Coverage

Before presenting our case study and its specific assumptions, the algorithms and their properties are presented for the general case with arbitrary cellular decompositions.

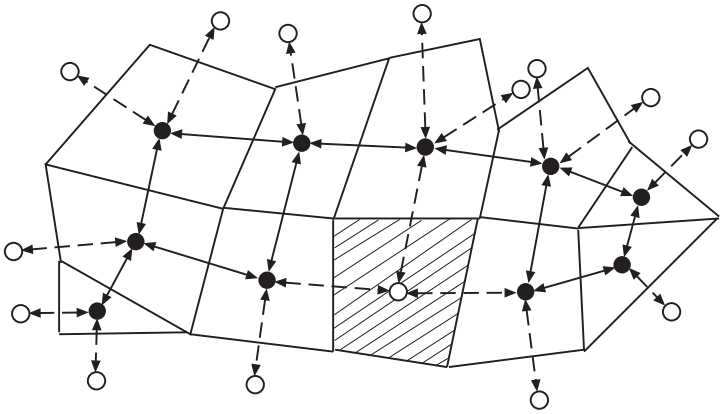

Fig. 1. An arbitrary environment with a cellular decomposition $\mathcal{V} \times \mathcal{E}$ consisting of a set of vertices $\mathcal{V}$ and a set of edges $\mathcal{E}$. Dashed edges can only be partially navigated and dashed cells are obstacles.

### *2.1 Non-Collaborative Coverage*

We describe the cellular decomposition of the environment as an undirected graph $G = (\mathcal{V}, \mathcal{E})$ with vertices $\mathcal{V}$ and edges $\mathcal{E}$ [10]. Each cell is represented by one vertex, and edges represent navigable routes between vertices and can be traversed by a robot in either direction. We denote $v \subseteq \mathcal{V}$ an individual vertex, and $\mathcal{E}^v = \{e_1^v, \ldots, e_{n_v}^v\} \subseteq \mathcal{E}$ the set of $n_v$ edges incident to $v$. Obstacles and borders are also represented in $\mathcal{V}$ with a vertex and are identified as non-navigable upon exploring the edge leading to it. We refer to the *neighborhood* $\Gamma(v)$ of $v$ as those vertices that are connected by an edge to $v$. We assume that the robot can determine all edges incident in $v$. A sample environment with an arbitrary cellular decomposition is depicted in Fig. 1.

For each time $t > 0$, a robot $i$'s state is described by $\mathcal{V}_t^i \times \mathcal{E}_t^i \subset \mathcal{V} \times \mathcal{E}$. Thus one robot's knowledge about its environment at a certain point in time is the subgraph $G_t^i = \mathcal{V}_t^i \times \mathcal{E}_t^i \subseteq G$. Thus, robots construct $G$ independently from each other.

After reaching and covering vertex $v$, assuming the average time to do so is $\tau^v$, the robot's state is updated with

$$\begin{aligned} \mathcal{V}_{t+\tau^v}^i &= \mathcal{V}_t^i \cup \{v\} \\ \mathcal{E}_{t+\tau^v}^i &= \mathcal{E}_t^i \cup \mathcal{E}^v. \end{aligned} \tag{1}$$

We define the set $DV_t^i \subseteq \mathcal{V}_t^i$ to be the discovered but not yet visited vertices as those vertices where one and only one vertex of a known edge is a visited

vertex, i.e.

$$DV_t^i = \{v \in \mathcal{V} \mid (v^*, v) \in \mathcal{E}_t^i \wedge v^* \in V_t^i \wedge v \notin V_t^i\} \tag{2}$$

where $(v^*, v)$ is the edge connecting $v^*$ and $v$.

Similarly, we define the set $RV_t^i(v) \subseteq \mathcal{V}_t^i$ to be the vertices that are reachable from vertex $v$ by the following recurrence relation:

$$(RV_t^i)_j(v) = \{v^* \in \mathcal{V} \mid (v, v^*) \in \mathcal{E}_t^i \wedge v \in (RV_t^i)_{j-1}(v)\} \tag{3}$$

with $(RV_t^i)_1(v) = \{v\}$. In other words, $(RV_t^i)_j(v)$ contains all vertices that are reachable from $v$ by at most $j$ edges.

Finally, the set of discovered, reachable and not visited vertices $DRV_t^i$ is defined by the intersection of the set of vertices $RV_t^i$ that are reachable from the current robot's location with the set of discovered, but not visited vertices $DV_t^i$:

$$DRV_t^i = RV_t^i \cap DV_t^i \tag{4}$$

To determine the next vertex to visit, the robot picks the closest vertex from the set of discovered, reachable and not visited vertices $DRV_t^i$ by solving

$$x = \arg \min_{x \in DRV_t^i} \boldsymbol{C}(v, x) \tag{5}$$

where $\boldsymbol{C}(v, x)$ is a cost function that is proportional to the time the robot needs to move between the centroids of the cells corresponding to the vertices $v$ to $x$. The value of $\boldsymbol{C}(v, x)$ is the solution to the *shortest-path on a graph problem* from $v$ to $x$ on $G$. Computationally, this requires to calculate the distance to all vertices in $DRV_t^i$ and then pick the vertex with the shortest distance. If there is more than one $x$ which minimizes $\boldsymbol{C}$, one of them is chosen at random.

For grid environments without obstacles (as in this paper), $\boldsymbol{C}(v, x)$ corresponds to the *Manhattan Distance*. In a plane with $v$ at $(v_1, v_2)$ and $x$ at $(x_2, y_2)$, $\boldsymbol{C}(v, x)$ is then given by $\|x_1 - x_2\| + \|v_1 - v_2\|$.

Finally, the best next vertex $v' \in \Gamma(v)$ to visit on the path toward $x$ is determined by once again determining the shortest path from $v$ to $x$.

We will now show that the algorithm leads to complete coverage if at least one robot does not fail.

**Proposition 1** Complete Coverage for a single robot: *Coverage is completed, when* $\min_{x \in DRV_t^i} \boldsymbol{C}(v, x) = \emptyset$ *(5), that is there are no discovered, unvisited vertices* ($DRV_t^i = \emptyset$) *and all vertices are reachable* ($RV_t^i(v) = V$, $\forall v \in V$), *i.e. the graph is fully connected.*

***PROOF.*** *Using (4), $DRV_t^i$ is an empty set, when all discovered vertices have been visited ($DV_t^i = \emptyset$). Using (2) this is the case if there exist no edge that connects an unvisited to a visited vertex. When there are no edges with unvisited vertices, $RV_t^i(v) = V, \quad \forall v \in V$.* $\square$

When working without explicit collaboration, it is possible that the environment is effectively covered *before* the individual robots complete, which is given by

$$\bigcup_{i \in \{1 \ldots N_0\}} \mathcal{V}^i_{t_{complete}} = V. \tag{6}$$

where $N_0$ are the total number of robots. In other words, even when each robot's $i$ coverage subgraph is only a subset of $V$, the *union* over all coverage sub-graphs might well comprise the entire graph. Throughout this paper, we are using the definition of $t_{complete}$ from (6) when referring to *time to completion.*

### *2.2 Collaborative Coverage*

Endowing robots with communication capabilities allows them to share their coverage map. Assume robot $j$ has visited a vertex. It will update its state $G_t^j$ according to (1) and then broadcast it. When robot $i$ receives the coverage map $G_t^j$ from robot $j$, it will update its own state by merging the two maps, that is

$$G^i_{t+\epsilon} = G^i_t \cup G^j_t \tag{7}$$

where $\epsilon$ is the necessary time for communication and information processing. Notice that a robot needs to evaluate (5) each time new information is available as the current goal vertex might already have been visited by another robot. Notice also that a robot shares his complete state and not the difference between his old and new state. This policy minimizes the potential loss of information due to communication faults, although it has clearly higher communication/energetic cost.

As robots do not arbitrate coverage paths beforehand, but only share information about actual coverage progress, completeness of collaborative coverage follows directly from the proof of completeness for the individual robot (Proposition 1).

### *2.3 Collaborative Coverage with Imperfect Localization*

On a miniature robotic platform there are two main causes of errors that lead to erroneous localization. First, imprecise actuators (e.g., wheel-slip) can

lead to a navigation failure when traveling from vertex to vertex in open-loop control. If a robot is dead-reckoning in the environment, such a failure might lead to a complete loss of positional information. Second, an imprecise sensor such as a global positioning system (GPS), e.g. provided by beacons/unique markers in the environment, can lead to an erroneous belief about the robot's whereabouts. As errors in localization potentially jeopardize the completeness property of any coverage algorithm, we propose an extension of our algorithm for environments where perfect localization cannot be guaranteed.

The basic insight is that as soon as perfect localization cannot be assumed, a robot has to associate a probability of actual coverage with each covered vertex. In order to increase this probability, the group of robots needs to repeatedly cover the whole environment, i.e. they need to do multiple *tours* or *patrol* in the environment. When robots do not communicate, this can be achieved by setting $\mathcal{V}^i_{t+\epsilon} = \emptyset$ as soon as $\mathcal{V}^i_t = \mathcal{V}$, which will start a new tour.

Using communication, we need to take into account that not all robots complete one tour at the same time (the time $\tau^v$ is an average value for the time to reach and cover one vertex, reflecting slip-noise and navigation errors). We associate with each vertex $v \in \mathcal{V}$ a tour index $\mathcal{T}^i(v)$. Upon visit of a vertex $v$, $\mathcal{T}^i(v)$ is increased by 1. Initially, $\forall v \in \mathcal{V}, \mathcal{T}^i(v) = 0$. Notice, that if the probability of incorrect localization is $p_f$, the likelihood of failing $\mathcal{T}^i(v)$ times is given by $p_f^{\mathcal{T}(v)}$ and thus the likelihood of the vertex $v$ being covered at least once is $1 - p_f^{\mathcal{T}(v)}$.

The set of vertices a robot should visit next is then given by

$$DV^i_t = \{v \in \mathcal{V} \mid (v^*, v) \in \mathcal{E}^i_t \wedge v^* \in V^i_t \wedge (v \notin V^i_t \vee \mathcal{T}^i(v) = \mathcal{T}_{min})\} \quad (8)$$

which will now direct a robot to visit a vertex with the lowest tour index, given by $\mathcal{T}^i_{min} = \min_{v \in \mathcal{V}^i_t} \mathcal{T}^i(v)$, or in other words, bias exploration toward areas with the lowest likelihood of completion. At the same time (8) ensures that robots will complete one tour before beginning the next. For all $v \in \mathcal{V}^j_t$, $\mathcal{T}^j(v)$ is broadcasted along the information mentioned in Section 2.2. Upon receipt, robot $i$ will additionally update its state as follows.

$$\mathcal{T}^i(v) = \max(\mathcal{T}^j(v), \mathcal{T}^i(v)) \quad (9)$$

### *2.4 Algorithm Properties*

*Communication complexity:* The collaborative coverage version of the algorithm requires $O(\|\mathcal{V}\|)$ broadcasts of state information: for each successfully covered vertex, the new robot state is broadcasted. These broadcasts are of

average size $O(\frac{N(N+1)}{2})$, where $N = \|\mathcal{V}\| + \|\mathcal{E}\|$. On the whole, the amount of data transferred is in the order of $O(\|\mathcal{V}\|^2)$.

*Memory requirement:* Immediately out of the definition of the robot state, the memory requirement of $O(N)$ follows.

*Algorithm complexity:* The complexity of the algorithm is dependent on the choice of the cost function $\boldsymbol{C}$. Assuming that $\boldsymbol{C} \approx O(\|\mathcal{V}\| \log \|\mathcal{V}\|)$, worst-case complexity is $O(\|\mathcal{V}\|^2)$, as $\boldsymbol{C}$ has to be calculated for each unvisited vertex once. However, typically, $\boldsymbol{C}$ needs to be evaluated only on a small subset of $\mathcal{V}$ in the neighborhood of the robot, thus decreasing the effective complexity.

For time to completion as well as for other metrics, we rely on the simulations described below to give practical results.

**Proposition 2** Probabilistic Completeness *Given a probability $p_f$ of failure of correct localization, $M \geq \lceil \frac{\ln \alpha}{\ln p_f} \rceil$ tours are required to achieve an average coverage level of $1 - \alpha$.*

***PROOF.*** *The probability of erroneously localizing itself in $M$ independent trials is given by $p_f^M$, which leads to the inequality $p_f^M \geq \alpha$ for the probability $\alpha$ to fail on an individual cell. Assuming $p_f$ to be constant over the whole environment, $\alpha$ can also be understood as the fraction of cells where localization failed.* □

For example, in order to achieve coverage of 95% given a sensor that provides correct localization with a probability of 70%, $M = 3$ tours are required. $M$ is an upper bound as our analysis does not take into account that each robot will potentially visit cells that have already been covered by one or more other robots while on the way to an unvisited cell. Due to this redundancy, which potentially makes up for failures of other robots, the effective coverage is much higher.

## 3 Case Study: Boundary Coverage of Regular Structures using a Swarm of Miniature Robots

Our case study is concerned with complete sensor coverage of the boundary of all elements in an environment. Inspection of the compressor section of a jet turbine engine is a motivating application [19]. The boundary coverage problem for structures with identical elements (Fig. 2) is equivalent to cover all *vertices* of a graph by a team of robots, which corresponds to coverage of

all cells of a grid when the elements are aligned in a regular pattern (as it is the case in [1, 2, 20]).

### *3.1 Experimental Setup and Robotic Platform*

A 60cm×65cm arena is populated with 25 blades in a regular pattern (Fig. 2 and Fig. 3), mimicking the rotor and stator blades of a jet turbine.

The Alice II robot is endowed with a PIC micro controller (368 bytes RAM, 8Kb FLASH), has a length of 22 mm, and a maximal speed of 4 cm/s. Four IR modules can serve as crude proximity sensors (up to 3 cm) and local communication devices (up to 6 cm). The response of the distance sensors is highly non-linear with a high resolution in close proximity (sub-millimeter) and a low resolution in the centimeter range (one increment in the sensor value corresponds to multiple millimeters). Robots are able to distinguish between blades, the arena boundaries, and other robots.

For inspection and localization, we use a VGA color CCD camera that is subsampled to 30x30 pixels [10]. The upper part of the blades are equipped with a unique color marker that consists of three colored horizontal bars (see Figure 2, right). Presence or absence of the 3 color channels (red, green, and blue) in the RGB camera image is used to encode 3 bits per color. Using the middle gray bar as reference (all channels at 50%) allows us to encode 64 different codes of which we are using 25 to identify each blade.

For communication and eventually transmitting sensory information to a base station, we developed a 2.4GHz radio module running TinyOS [10]. The radio also provides additional computational power and memory (8Mhz CPU, 4k RAM, 4MBit flash memory) and is used for executing the deliberative part of the coverage algorithm.

### *3.2 Reactive-Deliberative Robotic Controller*

Exploiting the regularity of the environment for navigation, the *Alice* can construct a graph with the blades as vertices, and possible routes between a blade and its 4-neighborhood as edges (Fig. 4, *right*, for an example graph). Edge traversal is achieved by a combination of dead reckoning and navigation along way points on a blade's boundary, (Fig. 4, *left*). At the same time, collisions are avoided reactively.

Way points can be distinguished by the robot's on-board sensors which can detect a blade's tip as well as measure the curvature of the blade using odome-

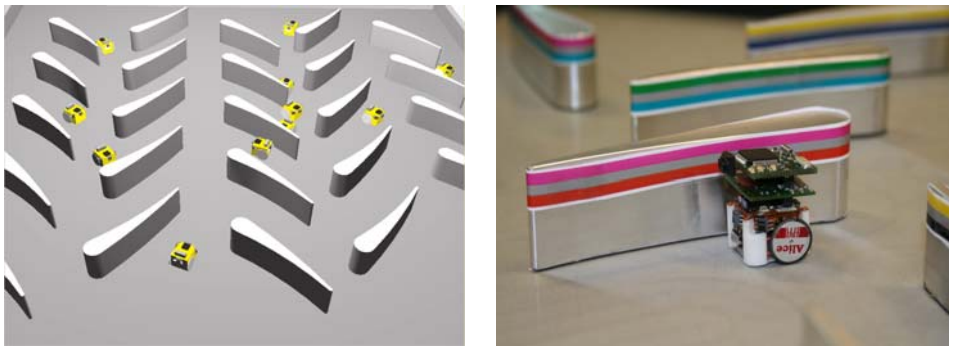

Fig. 2. *Left*: Overview of the turbine set-up in the realistic simulator. *Right*: Close-up on an Alice module equipped with a radio module in the real-world setup that is the motivation for the presented algorithms. Blades are equipped with a two-color code that serves for unique identification of a blade and therefore allows for the (discretized) localization in the environment.

try. For being able to read the barcode on a blade that is used for localization, the robot has to position itself so that the camera plane is parallel to the tangent of the blade's boundary. To achieve this, we implemented a closed-loop control algorithm that relies on distance measurements of the left and the right infra-red sensor. Near-optimal values for this controller as well as the near-optimal distance to the blade have been selected based on systematic experiments.

The feedback controller for aligning the robot in front of the blade deals well with the sensor non-linearities that are identical on the left and right side and has a high likelihood to succeed (see below for quantitative results for the success likelihood of the overall behavior including localization). Distinguishing the waypoints on the blade has been much harder despite the large difference in shape between sharp and round end. This is mainly due to the non-linearity of the distance sensor, which leads to drastical changes in sensor patterns for slight changes in distance to the blade. These changes are induced by wheel-slip that occurs when circumnavigating the blade using differential drive as well as varying amount of ambient light that offset the sensors.

Notice that although this experiment is very specific, the described implementation is an example for possible sensor-based navigation in miniature robots, which is prone to errors due to sensor noise or wheel-slip. In particular, sensor and actuator noise makes the execution time for the behaviors above non-deterministic and lead potentially to complete failures (see below).

Using the behaviors described above (refer to [10, 21] for more details), the graph is constructed on-line and systematically explored by the networked multi-robot coverage algorithm described above.


Fig. 3. The experimental setup endowed with colored markers that allow for unique identification of a blade using the on-board camera.

### *3.3 Microscopic Modeling and Simulation*

We quantitatively analyze the performance of the algorithm at two different model abstraction levels. Both models are microscopic, i.e. model the system at the level of an individual agent. On the lowest abstraction level, we simulate the environment and the robotic platform using a module-based realistic simulator *Webots* [22], Fig. 2, *left*. Webots is able to faithfully reproduce discrete intra-robot modules such as sensors, actuators, and transceivers for which the experimenter can specify dedicated nonlinear and potentially noisy responses. For instance, in all our simulations, Webots simulates wheel-slip of 10% if not otherwise noted.

We also implemented a Discrete Event System (DES) simulator of the model

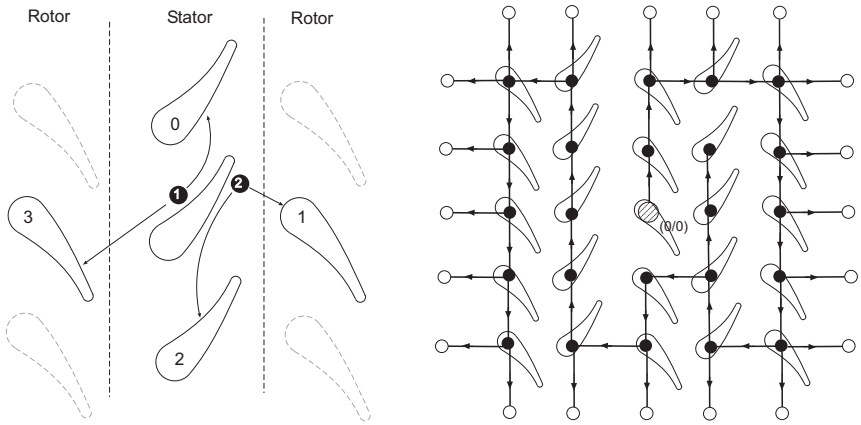


Fig. 4. *Left:* Way-points on a blade's boundary that can be navigated to using on-board sensors. *Right:* Possible trajectory for a single robot along a spanning-tree in a 5x5 blade environment (bold line). Backtracking paths are not shown.

described above using *MATLAB*, which corresponds to a microscopic agent-based simulator. The DES simulator models robots as points moving from vertex to vertex on the graph representing the environment. Multiple (body-less) robots can occupy one vertex, without interfering with each other. One simulation step consists in moving to a vertex with a certain probability of success, determining the vertex to visit next and potentially communicating state information. We calculated the histogram of the time needed for covering one vertex (including edge traversal) as well as the likelihood of successfully edge traversal for 6000 instances in Webots for different amounts of wheel-slip. The histogram of blade-to-blade navigation and coverage time as well as the relative likelihood for successfully traversing a certain number of edges are shown in Figure 5 for 10% of wheel-slip. We also performed measurements for 40% of wheel slip, which corresponds to that measured on the real miniature robot (not shown) [10].

Out of this distribution we randomly choose the time one simulation step is taking for one robot. Hence, results from the DES simulator are provided in seconds (rather than number of elements covered) and can be compared with the results of the Webots simulation.

On both abstraction levels, communication is simulated loss-less albeit for different ranges. In Webots, communication range is specified by the radius of an imaginary disc with a robot at its center, whereas in the DES simulator communication range is specified in terms of neighborhood relations on the graph (same vertex, 4-neighborhood, or global).

For the cost function $\boldsymbol{C}$, which is used at both modeling abstraction levels, we chose Dijkstra's algorithm with an uniform edge weight of 1. We also set $\boldsymbol{C}(v, x) = \infty$ if $x \notin RV_t^i$, thus (4) can be simplified to $DRV_t^i = DV_t^i$.

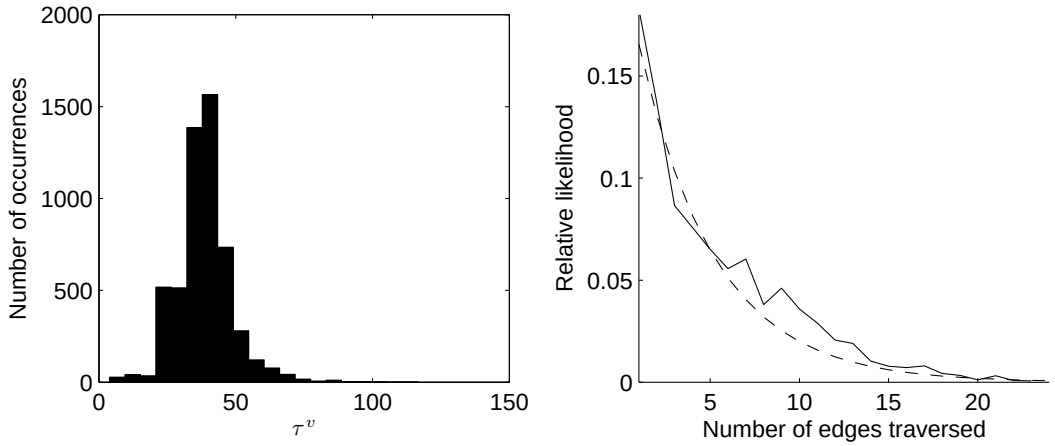


Fig. 5. *Left:* Histogram of times needed for blade-to-blade navigation including complete coverage ($\tau^v$). *Right:* Relative likelihood of successfully traversing 1 to 20 edges without failure. A geometric distribution (dashed) is super-imposed, which allows calculating $\pi^e$.

## 4 Results

In a first experiment, we show that the realistic simulation and the DES simulator produce quantitatively and qualitatively correct predictions for the same configuration (a configuration is defined by team size, environment size, communication range and reliability of localization) and investigate the limitations of the different modeling abstractions. We then rely on the DES simulator to explore a wider range of configurations. In particular, we vary the graph size and study the impact of varying communication range. We then compare the collaborative and the non-collaborative algorithm to a hypothetical optimal coverage algorithm. Then, we gradually decrease the reliability of localization and give results in terms of area covered and coverage time. We conducted 100 simulations for each configuration. As coverage performance shows a long-tail distribution, we provide the median time to completion and its 95% confidence interval for each experiment. Finally, results of 9 experiments with 5 Alice miniature robots are compared to predictions of DES and Webots simulations.

### *4.1 Matching of Modeling Abstraction Levels*

We first compare median coverage performance for 1 to 10 robots obtained by realistic simulation (Webots) and DES simulation (Fig. 6, *left*). We used the non-parametric Wilcoxon rank-sum test at 95% to determine if the qualitative match in medians seen in Fig. 6, *left* between data from realistic and DES simulation for different number of robots can be statistically verified. In other

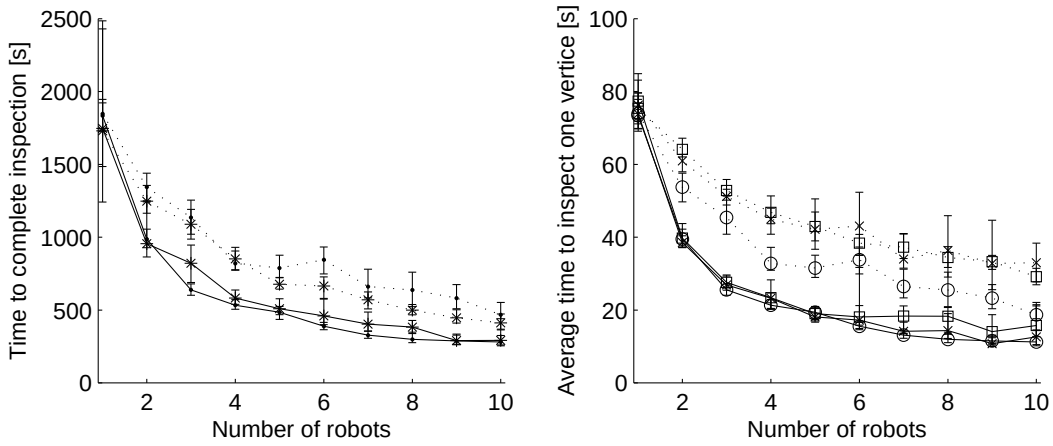


Fig. 6. *Left:* Time needed to achieve complete coverage for DES ($*$) and realistic simulation ($\cdot$) on a 5x5 square lattice, with (—) and without ($\cdots$) communication. *Right:* Time for average coverage of one vertex by the team (the time to complete coverage is normalized by the total number of vertices for environments of various size) in DES simulation with (—) and without ($\cdots$) communication for environments of 5x5 (o), 10x10 ($*$), and 15x15 ($\square$).

words, the probability that data from realistic and DES simulation have a different distribution is less than 5%. The agreement is better for smaller teams, which comes from the fact that DES simulation does not take into account collisions among the robots. This is an important observation showing the artifacts arising when comparing models at different abstraction levels.

### *4.2 Impact of the Environment Size*

In order to test the performance of our collaboration policy for different environment sizes, we used configurations which span two order of magnitude in terms of number of vertices: a 5x5 square lattice which corresponds to the real setup, as well as 10x10 and 16x16 square lattices. In terms of average time needed to inspect one vertex, robots show similar behavior on all environment sizes, for the smaller three configurations we show the results in Fig. 6, *right*. We see that when robots are not communicating, coverage progress (in terms of average time needed for covering a new vertex) is lower in larger environments. Thus, the larger the environment is, the more useful it is to let the robots communicate. This observation is confirmed by additional simulations on larger lattices (not shown).

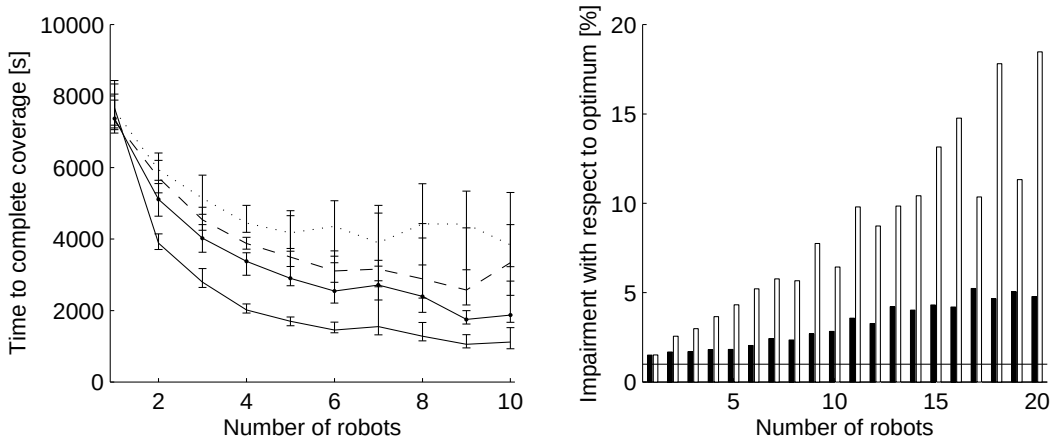


Fig. 7. *Left:* Time needed to achieve complete coverage with limited communication range: no communication ($\cdots$), same vertex communication ($- - -$), neighborhood communication ($-\cdot-$), global communication ($—$) (from top to down). *Right:* Comparison of the collaborative algorithm against optimal coverage on a 10x10 grid, 1 to 20 robots. The bar height represents how much worse our algorithm performs over a hypothetical optimal coverage algorithm compared to global communication (■) and no communication (□).

### *4.3 Communication Range*

As global communication is a strong assumption that can not always be guaranteed, we ran the DES simulator for four communication range configurations: no communication, robots communicate only when they are on the same vertex (range 0), robots communicate only with robots on a neighboring vertex (range 1), and global communication (infinite range). Time to completion for a 10x10 square lattice are depicted in Fig. 7, *left*. We observe that there is already considerable improvement in performance when communication is limited to robots being on the same or neighboring vertices.

### *4.4 Optimal Coverage*

We calculate a lower bound for the optimal coverage $T_{opt}$ as

$$T_{opt} = \frac{\|\mathcal{V}\|\tau^v}{N_0} \tag{10}$$

where $N_0$ are the total number of robots, and $\tau^v$ is the average time needed to cover one vertex. In Fig. 7, *right*, we explore the relative improvement an optimal coverage policy would give over the collaborative and non-collaborative algorithm for growing team sizes in a 10x10 environment from 1 to 20 robots. We observe that communication drastically increases the benefit of a larger

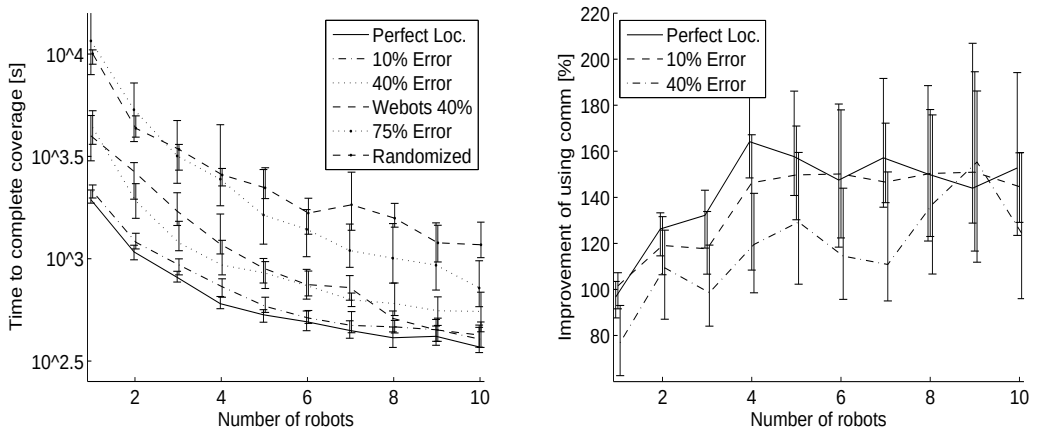


Fig. 8. *Left:* Median coverage time and its 95% confidence interval for global communication and different localization errors; comparison with randomized and Webots simulation (40% error only). *Right:* Improvement in terms of relative coverage completion time for having global communication over no communication.

team size, when compared with the same algorithm in which robots do not communicate.

### *4.5 Imperfect Localization*

We implemented the algorithm version with imperfect localization on the DES simulator for the 5x5 square lattice. At every vertex, a robot was provided with a random position with probability $p_f$, which corresponds to the probability of erroneous localization. We calibrated this value by taking 100 pictures per code from a blade labeled with the color code mentioned in Section 3.1, leading to erroneous localization in $p_f = 5.03\%$ of the cases. As these results do neither include possible misalignments of the robot to a blade nor the possibility of a robot trying to read the code from a blade's tip, this value of $p_f$ can be considered a lower bound in a real deployment. We thus ran simulations also for $p_f = 10\%, 20\%, 30\%$ and 40% in addition to all communication configurations described in Section 4.3. Notice that error in localization not only leads to sub-optimal path planning but also to wrong information shared with other robots. Time to completion for different $p_f$ as well as for a fully randomized approach are shown in Fig. 8, *left*. In the randomized version of the algorithm, robots do not coordinate but choose the next blade to visit at random with equal probability. Results comparing no-communication with global communication for different $p_f$ are shown in Fig 8, *right*. We see that when $p_f$ increases, the benefits of using communication get smaller. Up to a certain number, using more robots which communicate obviously improves the results.

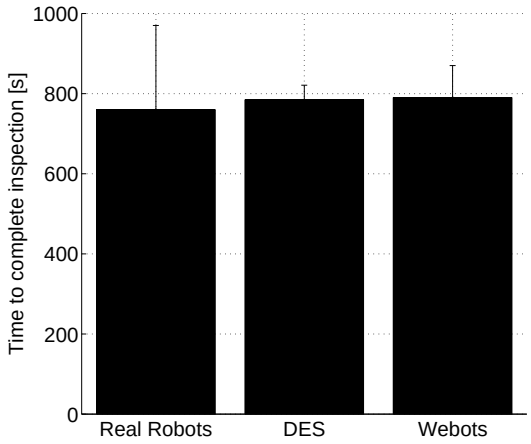


Fig. 9. Comparison of real robot experiments (left), DES simulation (middle), and realistic simulation (right) for 40% wheel-slip and $p_f = 33\%$. 100, 100, and 9 replications for Webots, DES, and real robots respectively.

In Fig. 8, *left*, we also show the results of a Webots simulation run when $p_f = 40\%$, which we can compare to the DES simulation configured at 40% error probability using the test described above in Section 4.1. Results are in good quantitative agreement when the number of robots is less than seven. We also compared the experiment where inspection is randomized and the configuration with 75% error probability. In a mostly 4-regular graph as in our case study, for less than 7 robots they lead to roughly equivalent outcomes (more than 95% significant), as choosing one out of 4 edges randomly is equivalent to a deliberative choice subject to 75% localization error.

In order to validate Proposition 2, we also measured coverage progress after every robot performed $M = 2$ tours (for $p_f = 10\%$ and $p_f = 20\%$), to $M = 3$ tours for $p_f = 30\%$ and to $M = 4$ tours for $p_f = 40\%$ error; the number of tours were calculated for a 95% coverage level (i.e. $\alpha = 0.05$). In all simulations, the average coverage was above 99% ($\alpha \leq 0.01$) confirming that the tour values calculated with Proposition 2 represent a conservative upper bound.

### *4.6 Real Robot results*

The algorithm has been implemented on a team of 5 miniature robots. Robots broadcast coverage maps periodically as defined by the algorithm. A static communication node was used to repeat the received coverage maps, which allowed considerable savings of energy on the individual robots as the radio could have been turned off when not used without missing messages. Result of 9 experiments are compared to DES and Webots simulations (100 replications) with 40% wheel-slip in Figure 9.

## 5 Discussion

The DES simulator, which numerically solves the recurrence equations (1) and (7) on the graph by sampling from the provided probability distributions, models the environment of our case study faithfully for the basic case where localization is perfect on the 5x5 square lattice (Fig. 6, *left*), as well as for 40% error in localization (Fig. 8, *left*). This allowed us to explore a wide range of other configurations on the computationally less demanding DES simulator.

There is improvement in time to completion even when the communication range is limited (Fig. 7, *left*). As robots keep on meeting, information propagates slowly to all robots. This is an important finding (see also [3]), as in most environments global communication cannot be guaranteed, either due to limitations on power consumption or due to environmental constraints that lead to communication loss; incidentally this is also the case for the miniature robotic platform and the case study being the motivation for the algorithms developed in this paper.

As the extension of the environment is unknown to the robots a priori, it is unlikely that the robots partition the environment optimally by chance. This is even more so when the number of robots is increased, as our algorithm does not perform any pro-active planning. Nevertheless, with increasing team size performance drastically benefits from communication (Fig. 7, *right*). The performance when using communication stays relatively close to the one of a optimal solution; without, it quickly worsens considerably. However, there is a point where adding more robots is no longer useful. In the 5x5 configurations, this point seems to be reached when there are 6 or more robots present (Fig. 7, *left*). This is the case regardless of the choice of communication range, different ranges will plateau at different average coverage times but for the same approximate number of robots.

We observe that using communication is still beneficial even with a large localization error of 40%, albeit its benefit gets relatively smaller (Fig. 8, *right*). Above 40% localization error the completion time comes close to that of a randomized approach (Fig.8, *left*). In this case a randomized approach might become competitive, in particular as it requires a less sophisticated robotic platform with less demand in power and computational resources and no need for communication. Then, a larger number of potentially smaller robots could achieve the same overall performance.

In all our experiments, randomized initial drop-off locations scattered over the whole area were chosen. We opted for this experimental setup, as we want to focus on the coordination aspect rather than a potential bias stemming from a particular initial configuration (e.g., central deployment vs. deployment

in a corner). We investigated the effect of the initial robot distribution in additional experiments (results not shown in this paper). We observe that whereas the performance for centralized deployments is generally lower than that of randomized ones due to navigation overhead generated by the crowded starting location, the algorithmic properties discussed above have shown to be persistent also in such scenarios.

## 6 Conclusion

We described an on-line algorithm for networked multi-robot coverage of environments with unknown extensions by a swarm of robots that can localize themselves, but are subject to significant sensor and actuator noise. The algorithm is formally described by a set of recurrence equations on a graph that are simulated using DES simulation, and validated in the realistic simulator *Webots*, which is known to faithfully reproduce real-world artifacts, and on a team of 5 miniature robots with a footprint smaller than one cubic inch without relying on any external computation or a central supervisor. Real robot experiments and both model abstraction levels provide good quantitative and qualitative agreement.

By carefully measuring sensor and actuator noise on the real platform and using realistic simulation, DES simulation allows us to explore various configurations, which model real-world use cases for various team sizes. In particular, we show that the proposed algorithm scales almost linearly with the number of robots employed when using communication. Thus, doubling the number of robots cuts the inspection time nearly in half, on the average, which is not the case when coverage is performed independently. We also show that even when communication and localization are strongly distorted, as it is the case during the real robot experiments, the algorithm stays robust and performance gracefully degrades to that of the non-collaborative or randomized algorithm, respectively. We could also demonstrate that it is possible to make probabilistic guarantees about the average coverage performance that explicitly take into account the limitations of an individual robotic platform in terms of positional noise.

## Acknowledgements

For performing this work, N. Correll and A. Martinoli have benefited of grants from the Swiss National Science Foundation (grant numbers PP002-68647 and PP002-116913). The authors would like to thank the anonymous reviewers for helpful comments.

## References


[1] X. Zheng, S. Jain, S. Koenig, D. Kempe, Multi-robot forest coverage, in: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Edmonton, Alberta, Canada, 2005, pp. 3852–3857.

[2] N. Hazon, F. Mieli, G. Kaminka, Towards robust on-line multi-robot coverage, in: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), Orlando, FL, USA, 2006, pp. 1710–1715.

[3] M. Jäger, B. Nebel, Dynamic decentralized area partitioning for cooperating cleaning robots, in: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), Washington, DC, USA, 2002, pp. 3577–3582.

[4] K. Easton, J. Burdick, A coverage algorithm for multi-robot boundary inspection, in: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), Barcelona, Spain, 2005, pp. 727–734.

[5] M. Batalin, G. Sukhatme, The design and analysis of an efficient local algorithm for coverage and exploration based on sensor network deployment, IEEE Transactions on Robotics 23 (4) (2007) 661–675.

[6] I. Wagner, M. Lindenbaum, A. Bruckstein, Distributed covering by ant-robots using evaporating traces, IEEE Transactions on Robotics and Automation 15 (5) (1999) 918–933.

[7] I. Rekleitis, G. Dudek, E. Milios, Multi-robot collaboration for robust exploration, Annals of Mathematics and Artificial Intelligence 31 (1–4) (2001) 7–40.

[8] H. Endres, W. Feiten, G. Lawitzky, Field test of a navigation system: autonomous cleaning in supermarkets, in: Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), Vol. 2, Leuven, Belgium, 1998, pp. 1779–1781.

[9] E. Acar, H. Choset, Y. Zhang, M. Schervish, Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods, Int. J. of Robotics Research 22 (7–8) (2003) 441–466.

[10] N. Correll, Coordination schemes for distributed boundary coverage with a swarm of miniature robots: Synthesis, analysis and experimental validation, Ph.D. thesis, Number 3919, École Polytechnique Fédérale Lausanne (2007).

[11] W. Burgard, M. Moors, C. Stachniss, F. Schneider, Coordinated multi-robot exploration, IEEE Transactions on Robotics 21 (3) (2005) 376–378.

[12] B. Yamauchi, Decentralized coordination for multirobot exploration, Robotics & Autonomous Systems (2–3) (1999) 111–118.

[13] M. Schwager, D. Rus, J. Slotine, Decentralized, adaptive coverage control for networked robots, Int. J. of Robotics Research. To appear (available online).

[14] H. Choset, Coverage for robotics—a survey of recent results, Annals of Mathematics and Artificial Intelligence 31 (2001) 113–126.

[15] Z. Butler, A. Rizzi, R. Hollis, Complete distributed coverage of rectilinear environments, in: Proc. of the Int. Workshop on Algorithmic Foundations of Robotics (WAFR), Boston, MA, USA, 2001.

[16] I. Rekleitis, A. New, H. Choset, Distributed coverage of unknown/unstructured environments by mobile sensor networks, in: A. C. Schultz, L. E. Parker, F. Schneider (Eds.), 3rd International NRL Workshop on Multi-Robot Systems, Kluwer, Washington, D.C., 2005, pp. pages 145–155.

[17] S. Thrun, D. Fox, W. Burgard, A probabilistic approach to concurrent mapping and localization for mobile robots, Autonomous Robots 5 (1998) 253–271.

[18] B. Gerkey, M. Matarić, A formal analysis and taxonomy of task allocation in multi-robot systems, Int. J. of Robotics Research 23 (9) (2004) 939–954.

[19] N. Correll, A. Martinoli, Towards multi-robot inspection of industrial machinery: From distributed coverage algorithms to experiments with miniature robotic swarms, IEEE Robotics & Automation Magazine. To appear.

[20] Y. Gabriely, E. Rimon, Spanning-tree based coverage of continuous areas by a mobile robot, Annals of Mathematics and Artificial Intelligence 31 (1–4) (2001) 77–98.

[21] N. Correll, S. Rutishauser, A. Martinoli, Comparing coordination schemes for miniature robotic swarms: A case study in boundary coverage of regular structures, in: Proc. of the Int. Symposium on Experimental Robotics, Rio de Janeiro, Brazil. Springer Tracts in Advanced Robotics, Vol. 38, 2008, pp. 471–480.

[22] O. Michel, Webots: Professional mobile robot simulation, Journal of Advanced Robotic Systems 1 (1) (2004) 39–42.