Edge Detection in Static and Dynamic Environments using Robot Swarms

Yara Khaluf
Department of Information Technology
Ghent University - imec, IDLab,
Ghent, Belgium
E-mail: yara.khaluf@ugent.be

***Abstract*—Robotic systems offer an attractive solution for a large spectrum of real-world applications that are hosted in dangerous or inaccessible areas for humans. In such applications, one of the fundamental tasks is to detect and mark particular features (e.g., pollution areas) in order to develop a proper response. In this paper, we focus on the specific problem of environmental edge detection using a swarm of homogeneous robots that can sense and act distributively using a large number of individuals. In our study, we consider both static and dynamic *2D* environments, in which the spatial distribution of the feature(s) may change over time. We verify our results using a set of physics-based simulations.**

I. Introduction

One of the robotics breakthrough in the last two decades is swarm robotics—communities of large number of simple robots that can perform complex tasks beyond the capabilities of their single individuals. The approach was inspired from natural swarms e.g., birds, fish, and social insects [1]. Swarm robotics has inherited a set of key advantages from its natural counterparts such as the high level of fault-tolerance due to its large number of robots. In addition to the high scalability of the system that results from relying on local communications in making their individual decisions which contribute to the emergence of the swarm global behavior. These advantages allow swarm robotics to provide a promising solution for a long list of applications ranging from cell-level (e.g., nano-robots) to space (e.g., robot swarms used for exploring life on other planets).

Missions in which robot swarms can be used include dangerous tasks for human, e.g., a field of harmful radiations or areas with high concentration of pollution. Another type of applications includes military missions such as isolating or guarding the borders of specific areas with different sensitivity levels against potential attacks. Likewise, robot swarms can offer an efficient solution in construction tasks across large-scale environments where environmental features may help as indication points to facilitate the construction process. For these missions and many others, robots need to be provided with the fundamental skills of what we refer to in this paper as the *environmental edge detection*. Environmental edge detection is the process of detecting and marking specific areas in unknown environments—in this paper, we consider only *2D* environments. These areas can be characterized by one feature or a combination of features, e.g., light, temperature, or radiation. As an example, we can consider a large environment in which some areas are characterized with a high concentration of CO_2. Drawing boundaries around those areas—referred to as environmental (CO_2) edge detection—may represents a necessary step to prevent humans from accessing those areas, and for cleaning up purposes. In case we were interested of isolating areas with, both, high concentration of CO_2 and high temperature, the detection will be adapted to take the combination of the two features into consideration. Without loss of generality, in this paper, we consider one environmental feature and we setup this to be the color of the environment's ground. Environmental edges in this case, represent parts of the ground where the intensity of the color changes sharply. A related field for detecting differences in color intensity is performed in the context of image processing and it is referred to as *image edge detection*. Our solution stays valid for detecting any other environmental features.

In our study, We use a swarm of N homogeneous robots which cooperate together to achieve a set of collective decisions concerning their detection of the environmental edges. We aim to achieve the following goals (i) to perform an efficient edge detection with the limited number of robots available, and (ii) to cope with any dynamic changes in the distribution of the environmental features. Achieving these goals using a robot swarm is challenging, since no global knowledge of the environment is available—robots rely only on their local communications and need to solve the problem distributively. Furthermore, robot swarms are limited physically and spatially—i.e. their performance drops when the number of robots increases beyond a specific threshold [2], [3].

The paper is organized as follows; Section II presents a list of works that focus on swarm robotics tasks, which are related to environmental scanning such as exploration and foraging with a brief discussion on the field of image edge detection. Section III describes our solution in which we characterize the robot's behavior on the microscopic (individual) level through its different phases. In Section IV, we introduce and validate through different sets of physics-based simulations the macroscopic behavior (i.e. the edge detection process) which emerges from the microscopic rules applied at the individual level. The paper is concluded in Section V.

II. Related Work

Robot swarms represent a promising solution for large-scale tasks due to, both, their large sizes and their flexibility [4]. A widely-used example is exploration, in which robots rely on a set of fundamental behaviors that help them spreading and dealing properly with tasks that are scattered across the unknown environment, e.g., to locate items that are of a particular interest [5], [6]. This behavior is mostly inspired from the well-known foraging behavior, which is observed in social colonies e.g., ants [7], [8]. Another fundamental behavior is to achieve a good coverage across large terrains. An efficient mechanism that is applied to realize this behavior is inspired by the physical process of gas expansion. The mechanism was applied in several studies of swarm robotics, for example, in [9], [10] to maximize the coverage of an unknown environment. The collective behaviors mentioned above are just examples of skills that are required by the robots while executing different missions such as under-water mission, for which robots need to explore large open environments and to locate environmental features [11]. Similarly, for in-body missions, in which robots need to work collectively in order to mark specific cells to which medicine should be delivered [12].

In nature, we observe species, e.g., social insects and animal groups, which respond collectively to environmental parameters such as light or temperature through distributing themselves spatially in a particular way according to the intensity of those parameters, [13]. This behavior is mostly referred to as aggregation. Nevertheless, this kind of aggregation is not the collective behavior we are investigating in this study. Our robots work to mark specific areas based on detecting a high concentration of one or more environmental features. In the aggregation—observed in nature—individuals gather at specific locations as a response to a particular stimulus [14]. Whereas, in our study, robots need to recognize the differences in the intensity of a perceived feature and to distribute spatially in a proper way to mark the edges that represent those differences in their highest values. As mentioned above, we have set the environmental feature to be the ground color of the environment. Hence, the nearest field that links to our study is image edge detection—a fundamental operation in the context of image processing, in which the goal is to detect places in the image where the brightness of the color changes sharply. There are several well-known edge detectors and some examples can be found in [15], [16]. The techniques used by such detectors include first filtering (e.g., Gaussian filters) for smoothing the image. Afterwards, they use some derivation functions such as the Laplacian or the first-order derivative function for recognizing changes in the color of the pixels. All operations and information that is needed for traditional edge detection such as smoothing, filtering, comparing to the exact neighboring pixels (i.e. the 8 neighbors), or computing global measures such as the maximum gradient require to deal with global knowledge of the image. Furthermore, even algorithms, which were developed following some collective intelligence mechanism, e.g., in [17], were relying on global knowledge as well such as the image threshold. In summary, both, the techniques and the global knowledge exploited in the traditional image edge detection make it not feasible to import these to the robots for performing environmental edge detection. The only technique that can be imported safely is *thresholding*[1]. Applying the threshold technique helps detecting environmental edges regardless their degree of sharpness—the rate at which the spatial distribution of the feature changes. In other words, even in scenarios where the spatial distribution of the feature changes with small steps (i.e. the edges don't represent a sharp change) such as "gas edges", the threshold technique facilitates taking the binary decision, whether the change represents an edge or not. For robot swarms, thresholds are computed collectively without any global knowledge.

III. Microscopic Behaviors for Environmental Edge Detection

In this section, we describe the behavior of the individual robots, which is designed to detect the edges of specific environmental features. Environmental edge detection is performed distributively relying *only* on the robots' local information—both the perceived information and the information received from the local neighbors. This imposes a set of challenges which we are addressing in the proposed solution. In this paper, we distinguish between two types of environments; (i) static environment and (ii) dynamic environment. In static environments, the edges don't change over time. While in dynamic environments, the spatial distribution of the feature may change over time, hence the robots will need to re-mark the new edges.

A. Adaptive threshold phase

Thresholding mechanism is used mostly in the traditional edge detection to identify whether a particular change in the color intensity represents a true edge—i.e. when the difference is greater than the pre-defined threshold. The threshold can be computed using different algorithms, all of those use global information about the image. In our solution, we similarly rely on the threshold mechanism for defining the edges of the environmental feature, however, we differently do so in unknown environments. Since no global knowledge is available for such environments, providing the robots with a global appropriate threshold that reflects properly the feature change in the environment is not feasible. At the same time, the success in detecting true edges depends to a large extend on the use of an appropriate threshold. Trying to overcome this challenge by providing the robots with a user designed threshold will probably result in an undesirable edge detection, since using a high threshold may lead to miss true edges and using a low one may lead to select false edges. Therefore, this threshold needs to be computed by the robots distributively and over a sufficient time to scan the environment throughly.

[1] A technique used to identify true edges by comparing the change in their color intensity to some particular threshold.

Computing local thresholds—i.e. robots' own thresholds—using a few measures performed by the robots shortly after starting the task will reflect insufficient information that is not enough to represent the average change in the intensity of the feature across the environment. Thus, in our solution, the robots converge on a proper threshold (i.e. a global averaged value) only after exploring the environment and collecting enough samples (observations). This task is performed by the robots during the first phase of the solution which is referred to as the ***adaptive threshold phase***.

In this phase, each robot starts in the "**exploring**" state. In this state, the robot wanders in the environment to sample enough observations. An observation is any change in the intensity of the environmental feature. For the rest of the paper we use only the ground color as the feature we are focusing on in our experiments, and as an example of other environmental features. As soon as, an observation is made or received from the neighborhood, the robot updates its local threshold. This update is an aperiodic operation that is triggered whenever a new observation is obtained and the robot switches to the state "**update observation table**". In this state, the robot updates a local table that holds a history of all obtained observations by adding the new observation. This table is used to compute the robot's local threshold by each robot individually. According to the size of the robot's memory, previous observations are kept and removed when needed by their time of creation.

As mentioned above, the threshold is used in the traditional edge detection for images to indicate true edges by comparing the threshold with the slop of the zero-crossing in the colors. In the case of environmental edge detection, this threshold is used to recognize potential edges from being compared to the observations performed by the robots and which represent the differences perceived in the intensity of ground color. An observation occurs when the difference between any two values of the sensed color is larger than zero. We define the robots' observations as a triple, $O(i,j,k)$, where j is feature (color) for which the observation is made, by robot i at time step k. This observation is defined as the maximum difference between the values perceived of feature j at the time step k using all the sensor inputs:

$$O(i,j,k) = max|O_l(i,j,k) - O_m(i,j,k)|, \qquad (1)$$

where $O_l(i,j,k)$ and $O_m(i,j,k)$ are the observations obtained from the sensors l and m, respectively. $(l,m) =^2 P_w$, where w is the set of sensors used to perceive the environmental feature j (i.e. color sensors), and ^2P_w are the permutations in pairs of all the sensors in w. Figure 1 illustrates an example of how a robot that is equipped with 4 feature sensors compares the data perceived from these sensors in order to compute the value of its observation as in Eq. (1).

Each time a new observation is obtained or received, the robot updates its local threshold as in the following:

$$\theta_i(t) = (1-\alpha)\frac{\sum_{M_i} f_{O(i,j)} \times O(i,j)}{M_i} + C, \qquad (2)$$

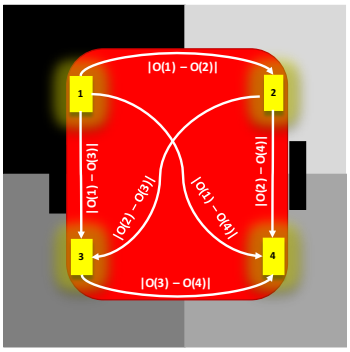


Fig. 1: An illustration of how a robot with 4 sensors for the environmental feature compares the perceived data in order to compute the value of it observation.

where $\theta_i(t)$ is the local threshold of robot i computed at time t, $O(i,j)$ is an observation of the feature j, that is either made by robot i or by one of its neighbors, $f_{O(i,j)}$ is the frequency of the particular observation $O(i,j)$—i.e. the number of times this specific observation was obtained up to the time point t. (Repetitive observations at different time steps are considered as one observation with > 1 frequency). M_i is the total number of observations made or received by robot i up to the time point t. $\alpha \in [0,1[$ and $C > 0$ are design parameters that control the relation between the computed threshold and the average change in the intensity of the environmental feature. Setting α to a high value or/and C to a low value results in lowering the robot's local threshold below the average of differences in the intensity of the feature. This will allow the robots to detect and mark weak edges, which can be useful in detecting dangerous environmental features, where even the areas with low concentration of the feature are required to be detected and isolated. The main restrictions associated with this approach are related to the size of the swarm which may limit the number of edges possible to mark. In addition the physical constraints that are common in robot swarms such as the influence of spatial interferences on the performance of the swarm. Finally, lowering the threshold has the well-known side effect of increasing the probability of detecting and marking false edges. Marking false edges in traditional image edge detection is not desired from the correctness point of view. However, in our system, in addition to the condition of correctness, marking false edges leads to losing robots, which is a limited resource. In general, the selection of both α and C can be performed as a function of (i) the number of available robots, and (ii) the level of detection required for the specific environmental feature(s).

Using a collective robotic system to converge distributively on a proper threshold includes a set of robotic-related challenges, In the following we list the main challenges, which we needed to overcome during our study:

- **Robots' spatial distribution vs. the edges' spatial distribution:** in order to obtain a proper threshold that reflects the average change in the feature's intensity, the

coverage of the environment needs to be maximized to allow the robots to maximize the range from which they sample their observations. For coping with this challenge, we applied the well-known mechanism of gas diffusion for spreading the robots across the environment during their motion. This mechanism facilities a better exploration with a maximized coverage [18].
We allow robots to start their task from a uniform spatial distribution. Uniform distribution is used in order to synchronize the best with the process of traditional edge detection (where any pixel in the image can be considered from the starting point of the process). In general, robots could also start from a particular location (e.g., nest). This will not influence the logic of the solution. It may just prolong the time until a proper threshold is adapted. By using the uniform distribution, the robots' initial observations will reflect the average changes in the feature's intensity across the whole environment. However, since edges are located at positions with a remarkable difference in the intensity, robots need to explore and sample beyond the observations obtained at their starting positions. This will allow their local thresholds to change until they stabilize around an average value that reflects a proper threshold.

- **Sharing the observations with the neighborhood:** when a new observation is made, this observation is broadcast to the robot's neighborhood. The broadcast of the recent observation continues until a new observation is obtained. This continuous broadcasting makes use of the mobility of the system to help spreading the obtained observation to the largest number of robots while moving around. Motion provides the robots with a dynamic neighborhood list, in which the neighbors of a robot change during its motion. This allows the robot to convey its observations to different neighbors, and since robots in swarms rely *only* on their local communications, maximizing the number of neighbors who receives a particular message can be only achieved by prolonging the transmission period of that message. Nevertheless, the challenge here emerges from the significant difference between the communication speed and the motion speed. Because of this difference, the robot may receive the same observation from the same sender multiple times before the robots get out of the communication range of each other. This may bias the computation of their local thresholds—amplifying the weight of a particular observation. In order to solve this problem, the robot broadcasts both its ID and a time stamp associated with that specific observation. The receiver, in this case, uses this information to identify whether this observation is a new one. In case the observation is new, the local threshold is updated, otherwise the receiver ignores the message.

The robots continue to explore their environment and to update their local thresholds until the end of this phase is reached. This end is decided by each robot individually, when the change in the computed threshold stabilizes for a given period of time Δt_θ. This change is measured by the robot using the average deviation around the mean of the local thresholds computed by that robot over the time period Δt_θ. As soon the average deviation becomes smaller than a given error θ_{err}, the robot decides to end adapting its local threshold. The stopping condition is given by:

$$\frac{\sum_{k=1}^{M_i(\Delta t_\theta)} |\theta_i(k) - \hat{\theta_i}|}{M_i(\Delta t_\theta)} \leq \theta_{err}, \tag{3}$$

where $M_i(\Delta t_\theta)$ is the number of thresholds which were updated by robot i within the time period Δt_θ, $\theta_i(k)$ is the $k-th$ threshold computed by robot i within the time period Δt_θ, and $\hat{\theta_i}$ is the mean local threshold computed by robot i. The behavior of the individual robots in the ***adaptive threshold phase*** is described in Algorithm 1.

Algorithm 1: The algorithm followed by the individual robots in our physics-based simulations during the adaptive threshold phase.

```
while Not(End the adaptive threshold phase) do
    Explore the environment using diffusion;
    for any made or received observation O(i,j,t) do
        if O(i,j,t) is a new observation then
            Update the local table of observations;
            Re-compute the local threshold;
        else
            Ignore the observation;
        if O(i,j,t) is made by the robot then
            while no new observation is made yet do
                Broadcast O(i,j,t) + robot ID + time
                  stamp;
if Eq. (3) returns true then
    End the adaptive threshold phase
```

B. Edge detection phase

The second phase in our solution is dedicated to the process of finding and marking the edges of the environmental feature (i.e. the ground color). This phase is referred to as the ***edge detection phase***. Edge detection is performed based on the robots' local thresholds, which were computed in the previous phase. In this phase, each robot starts in the state "**edge_searching**". In this state, the robots wander in the environment, applying the diffusion mechanism for maximizing the coverage of the environment and visiting as many edges as possible. While moving around, robots measure the changes occurring in the intensity of the environmental feature as explained in Section III-A. In case any of these observations,

e.g., $O(i, j, t)$—perceived by robot i at time t according to Eq. (1)—has a higher value than the robot's local threshold θ_i, the robot marks the location at which this observation was made as an edge. This is performed by simply stopping at the location of the particular observation. After marking the location, the robot switches its state to the "**edge_marking**" state.

While in the state "edge_marking", the robot continues to perceive the intensity of the environmental feature at its location, however, with a lower frequency than the frequency applied when the robot was in the "edge_searching" state. This continuous assessing of the feature's intensity at the marking location allows the robot to capture any change in the spatial distribution of the environmental feature. In case, the feature's intensity changes, the robot switches back to the adaptive threshold phase, and starts exploring its environment for computing its new local threshold. Additionally, this robot starts to broadcast messages to its neighborhood—for a limited time—to inform it about the change in the spatial distribution of the environmental feature. Robots which receive this message, even if they still don't sense the change, they switch similarly to the adaptive threshold phase. The message keeps spreading by all robots which switched to the adaptive threshold phase, and reaches the maximum number of other robots thanks to the robots' motion.

This above-described behavior allows robots to cope with dynamic environments, in which changes in the feature can occur over time and hence, require the robots to redistribute automatically. The behavior of the individual robots, in both phases, is captured by the state machine shown in Figure 2.

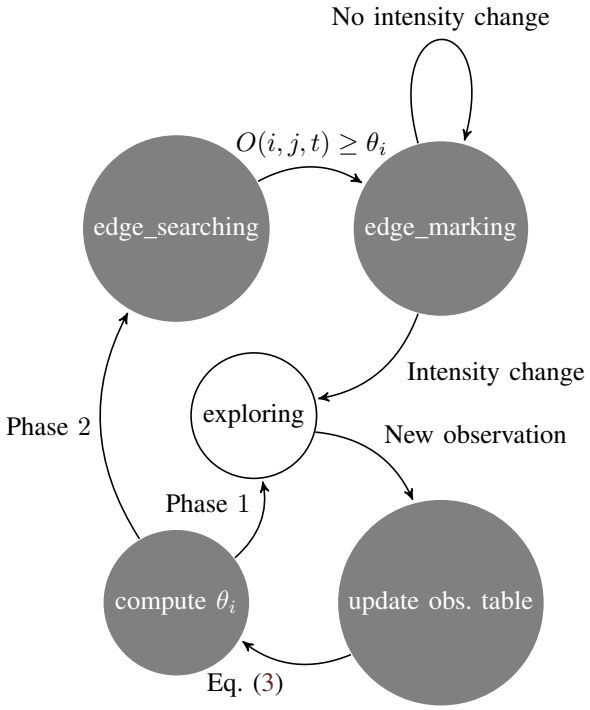


Fig. 2: The state machine used by the individual robots in both phases of the edge detection solution. The start state is "exploring".

Using a collective robotic system to search and mark edges includes a set of robotic-related challenges. In the following we list the main challenges, which we needed to overcome during our study:

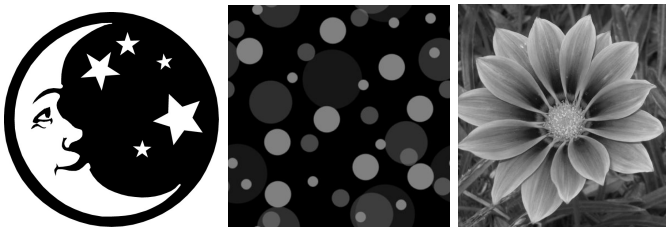
- **Building closed areas and block searching robots:** when robots start to mark edges, they stop as soon as the change in the intensity of the feature exceeds their local thresholds. This behavior can lead to build up closed areas, in which robots are spatially arranged in closed shapes. This can block other searching robots that happen to be in the closed shapes when the edges were marked. The solution we have adapted to overcome this challenge, was to allow the robot which is in the "edge_searching" state to notify it neighbors in the "edge_marking" state of its need to pass through. As a response to this message, the robot in the "edge_marking" state preforms a limited motion in its most-free direction—i.e. the direction in which the obstacle avoidance sensors used by the robot (e.g., infrared sensors) report the lowermost measures associated with the most free space to move through. Afterwards, it waits for the blocked robot to escape before it reverts its motion direction back and return to mark its edge. Robots use logical flags to avoid switching to the "edge_marking" state while passing the edge which was marked by the other robot. It is designed this way to avoid exchanging the observation tables between the two robots and additional messages to switch roles. This solution has proved its efficiency in most cases, however, there are still some scenarios in which getting the blocked robots free is a none-trivial task. Examples include cases where moving the marking robot doesn't provide an appropriate exist for the blocked robot.
- **Limited number of robots:** albeit swarm robotics is characterized with its large sizes, however, robots are still a limited resource. This limitation influences the capability of the system to mark the environmental edges in case those need more robots than available. One possible solution that may help in balancing the number of available robots with the density of edges in the unknown environment would be through adapting the distance between the neighboring robots, which are marking the same edge. This should be performed so that the minimum number of robots (but enough) mark the edge. The additional robots can then be used in marking the other edges.

IV. The Macroscopic Performance of the Environmental Edge Detection

In this section, we evaluate our solution including its two phases by investigating the macroscopic behavior of the swarm, i.e (i) the convergence on a proper threshold by all robots that allows them to perform an efficient edge detection, and (ii) the proper detection and marking of the feature's edges. We test the macroscopic behavior using a

set of physics-based simulations performed using the state-of-the-art simulator for swarm robotics, *ARGoS* [19]. ARGoS allows to simulate a large number of robots efficiently while taking the detailed characteristics of the robots' physics into consideration. In our experiments, we use *Footbots*[2] to build our homogeneous swarms. We split our experiments in the following parts; First, we launch experiments to verify the efficiency of our solution in converging on a global average threshold (GAT). The second set of experiments is dedicated to illustrate the efficiency of our swarm in detecting and marking edges using the thresholds computed in the first phase. The last set of our experiments is used to validate the efficiency of our solution in coping with dynamic environments.

In our experiments, we use three images for the environment's ground (i) the moon ground, (ii) the bubbles ground, and (iii) the flower ground, see Figure 3. We assume that areas, which need to be isolated (e.g., hazardous areas) by marking their edges, are the bright (white) areas in the images. The need to isolate such areas decreases gradually with the color of the ground getting darker (black areas don't need to be isolated). We use an arena of 8×8 m^2 and we experiment with different swarm sizes.



(a) (b) (c)

Fig. 3: The three grounds used in our physics-based simulations. (a) the moon ground, (b) the bubbles ground, and (c) the flower ground.

A. Results of the adaptive threshold phase

In this set of experiments, we aim to validate (i) the efficiency of our solution in allowing the swarm to converge on a proper threshold. (ii) How this convergence changes according to, both, the complexity of the ground—the density of edges in the image—and the size of the swarm. We have set, both, α and C in Eq. (2) to zero for all the experiments in this set, since these two parameters have their influence when robots start to mark edges (in the second phase) and not during the computation of the adaptive threshold. Therefore, their values will be adapted for the experiments dedicated to the edge detection in Section IV-B. Additionally, θ_{err} in Eq. (3) is set to 5%. Figure 4 depicts the process of converging on a global average threshold (GAT) in the swarm. On the left side of the figure, we can see the progress of GAT—i.e. the average of the robots' local thresholds—over time. Whereas, on the right side of the figure, we can see the local thresholds computed by the robots throughout the time of the experiment. The experiments stop when all robots have managed to mark edges, or when a specific termination time is reached (this was set to 3000 time step). The threshold computed for the moon ground is illustrated in Figures 4a,b, for the swarm sizes 250 and 150, respectively. First, we can notice that for both sizes, the swarm is reaching the same global average threshold (GAT). Secondly, we can notice that smaller swarms need longer time to converge on their GAT. Figures 4c,d, show the same results for the bubbles ground, and Figures 4e,f, for the flower ground. Similarly for these grounds, our conclusions stay valid. Our results were collected over 35 runs for each configuration; i.e. environment ground and swarm size, hence, 210 total runs.

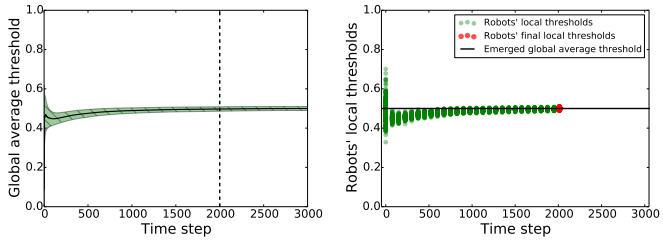
A further remark is related to the fast convergence on the final GAT. This can be explained due to the uniform distribution of the robots across the environment with association to the spatial distribution of the edges, which is more or less equally distributed across the environment. This allows the average of the robots' local thresholds (GAT) to stabilize around a proper value that reflects the changes in the intensity of the ground color in a *short time*. The further updates of the GAT after the big convergence jump and before stabilization are due to the contributions of the observations made by the robots at locations which were not visited yet. The individual contributions of such observations result in the small updates of the GAT until it stabilizes and the adaptive threshold phase is terminated.

We analyze the results from the ground complexity point of view. For this purpose, we have plotted in Figure 5 the convergence time of GAT for the three grounds we are using in our experiments. For all grounds, we can notice that the convergence time increases with increasing the complexity of the ground—here the ground complexity is a measure of the edge density in the image used for the ground. For example, the flower ground has a higher density of edges than the moon ground, hence the convergence time of GAT on the flower ground is longer than that one on the moon ground.

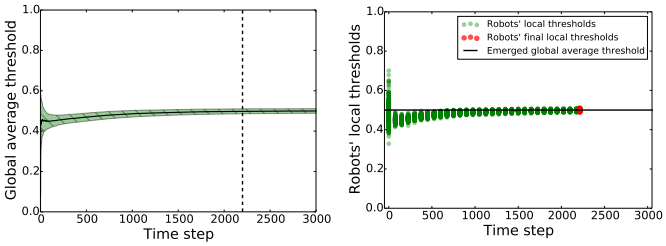
B. Results of the edge detection phase

In this set of experiments, we aim to validate (i) the efficiency of our solution at leading the swarm to perform a proper edge detection, in addition to (ii) validate the adaptivity of the detection level in relation to the robots' local thresholds. Using the color feature, as our environmental feature, has helped us in evaluating the quality of the edge detection performed by the swarm. This is due to our a priori knowledge of the edges in the images used for the ground. Hence, we could compare the edges marked by the robots with the true edges in those images and verify to which degree the level of detection matches with the edges of the images. The level of detection refers to the level of intensity in detecting edges—from the detection of the strongest edges to the detection of the weakest ones. This measure is evaluated in our experiments as a function of the robots' local thresholds computed in Eq. (2).

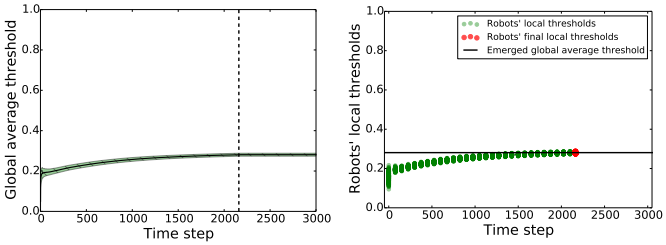
[2]Footbot is a wheeled robot which is used widely in the research of swarm robotics, you can check it out at www.swarmanoid.org/swarmanoid_hardware.php.html



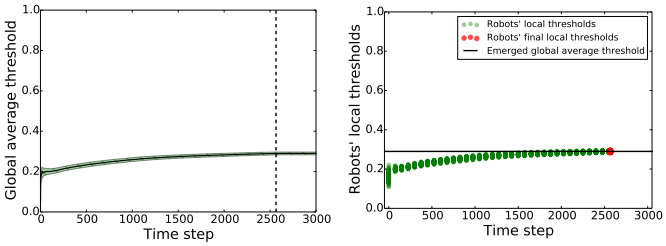
(a) 250 robots on the moon ground.



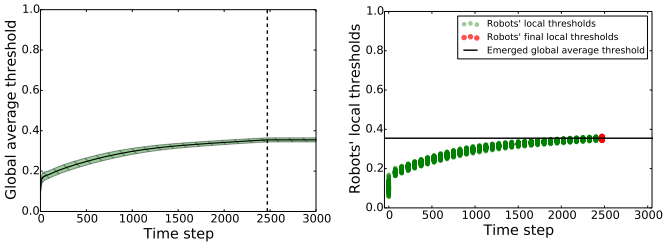
(b) 150 robots on the moon ground.



(c) 250 robots on the bubbles ground.



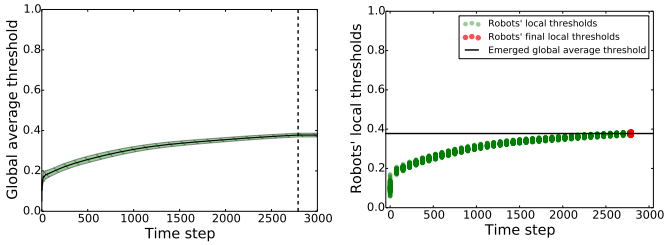
(d) 150 robots on the bubbles ground.



(e) 250 robots on the flower ground.

Figure 6 shows the detected edges using a swarm of 300 robots on the three ground images[3]. The robots, in this set of experiments, have computed their local thresholds with setting both α and C in Eq. (2) to zero and θ_{err} in Eq. (3) to 5%. In some trails of these experiments, all robots have succeeded in detecting edges and have stopped for marking them before

[3]The Footbot robot which we are simulating with has a diameter of 17 cm, hence, our arena of 8×8 m^2 can contain up to 1936 Footbot



(f) 150 robots on the flower ground.

Fig. 4: The global average threshold (GAT) computed over the three different grounds using swarms of, both, 250 and 150 robots.

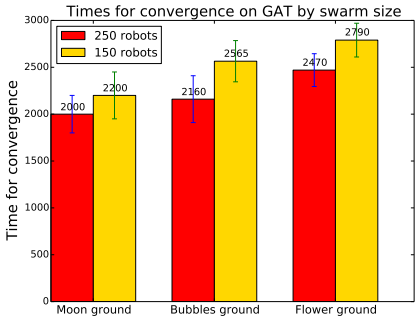


Fig. 5: The convergence time of the global average threshold (GAT) over the three grounds using two swarm sizes, 250 and 150 robots.

the experiment has terminated. Whereas in most of the trails, there was a few robots, which were still searching for edges, when the time of the experiment has reached its end (here 3000 time step). In the left column of Figure 6, we can see the final spatial distribution of the robots over their detected edges. The same robots' distribution is shown in the right column of Figure 6, however on a black ground for helping to obtain a better recognition of the image, whose edges are marked by robots. In all the cases, we can notice a high degree of matching between the original image and the image evolving from the edges detected by the robots.

As mentioned above, the robot's local threshold can be tuned using either the parameter α or the parameter C in Eq. (2). By tuning the robot's local threshold, the level of edge detection changes as well. In order to validate this, we use three variants of the parameter α while keeping the parameter C not changed. The bubble ground is used in this test with the following settings (i) $\alpha = 0.9$, (ii) $\alpha = 0.5$, and (iii) $\alpha = 0$. We check accordingly the level of edge detection that was reached by a swarm of 300 robots.

Figure 7 illustrates the different detection levels that can be reached with the three settings of α. A high α leads to decrease the robot's local threshold, and therefore allows that robot to detect weaker changes in the intensity of the color as edges. This what we can notice in Figure 7a, where even the edges

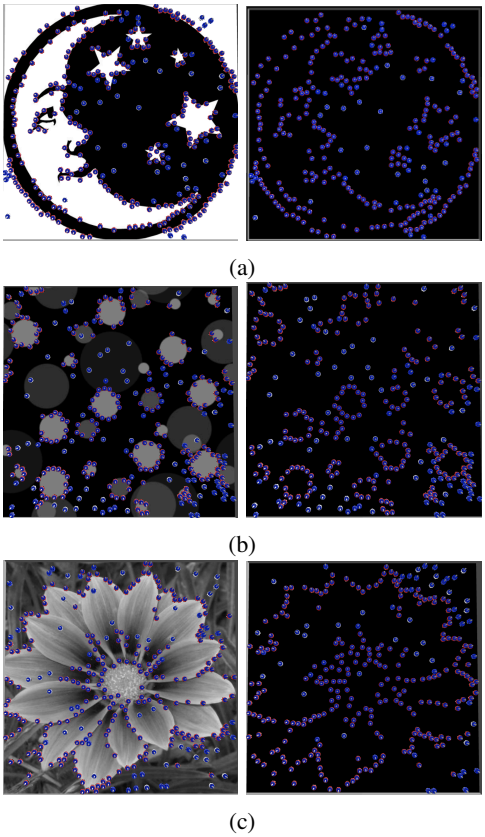


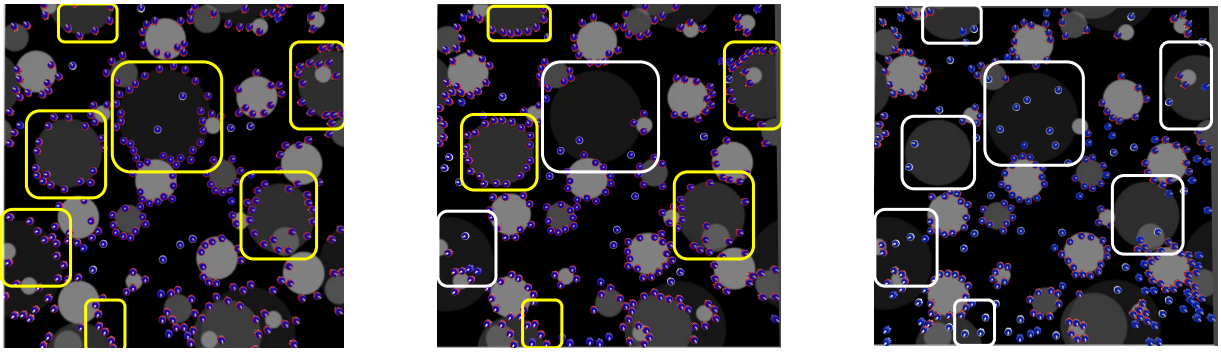
Fig. 6: Edge detection using a swarm of 300 robots on the three different grounds. (a) the moon ground, (b) the bubble ground, and (c) flower ground.

of the very dark bubbles were marked by the robots (yellow rectangles are used in Figure 7a to highlight the dark edges which were marked by the robots). While decreasing α to 0.5, allows robots to skip the weakest edges, notice in Figure 7b, white rectangles are used to highlight the edges that robot have missed in comparison to the edges marked in Figure 7a. The smallest number of edges, however, the strongest ones, are marked when α was set to 0. Those are the edges of the bubbles with the lightest colors. We can observe this results in Figure 7c, in which robots have missed all dark edges which were detected for higher values of α. White rectangles are used again to highlight the edges that robot have missed in comparison to the edges marked in Figure 7a and Figure 7b.

Additionally, we have implemented the percentage of the correctly detected pixels by the robots to the total number of pixels in true edges—as a quantitative measure—that are detected following the well-known Laplacian detector. This measure can be denoted as P_{TC}. We have tested using the bubbles ground image and repeat our experiments over 30 runs. For all values of α(s), the detected pixels were all correctly belonging to true edges. For $\alpha = 0.9$, we had $P_{TC} = 0.63$, for $\alpha = 0.5$, we had $P_{TC} = 0.57$, and for $\alpha = 0$, we had $P_{TC} = 0.51$. Using more robots may increase this measure.

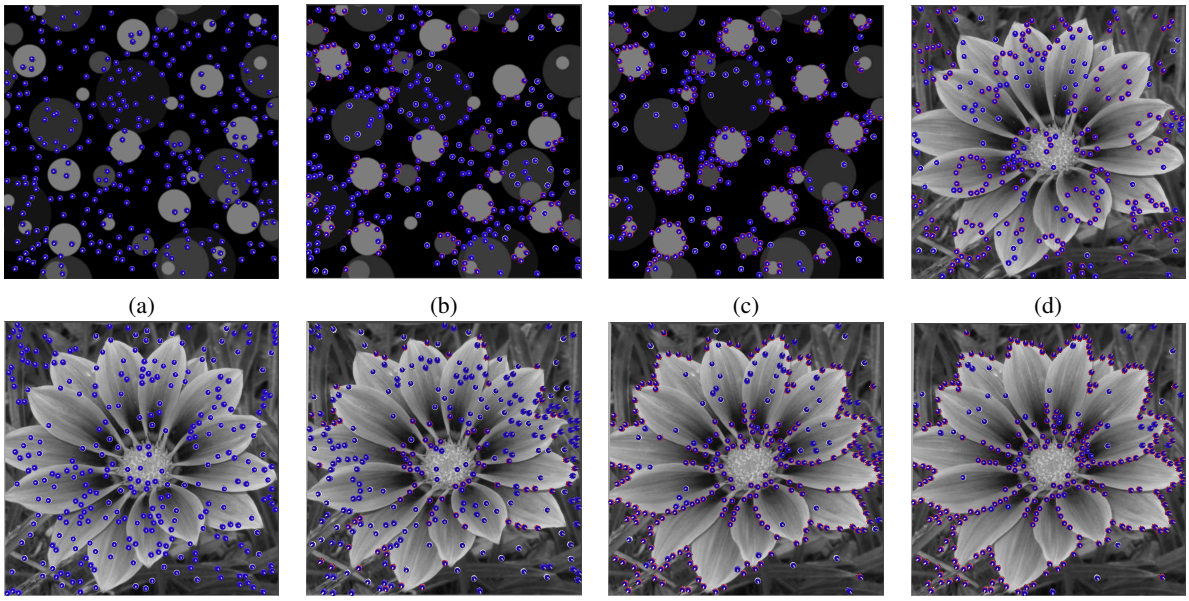
C. Results in dynamic environment

The last set of experiments is performed to validate the efficiency of our solution in coping with dynamic environments. The changes in such environments are modeled as a set of discrete consecutive events of change in the spatial distribution of the environmental feature. For ground images, this is simulated as a sequence of images which change over time. Without lose of generality, we limited the sequence of images, in this set of experiments, to two in the first experiment and three in the second one. Using a larger set of images to allow simulating a smoother change in the distribution of the feature can be handled by a repetitive execution of our solution. However, this would be a time-consuming process that will provide no added value to the validation of the approach. We similarly use a swarm of 300 robots and for the first experiment two of our grounds; the bubbles ground and the flower ground. We start by distributing the 300 robots uniformly over the bubble ground. We launch the experiment and allow robots to compute their local thresholds and to search for edges on the bubbles ground and mark those. Robots which are marking edges will keep sensing the ground over relatively long time periods (i.e. with a lower frequency than the one used when robots were still exploring the environment). At the time step 2000, we changed the ground to the flower one. Robots which sense a change in the intensity of the color at the edges, which they are marking, switch to the adaptive threshold phase and start exploring their environment. These robots start to broadcast a message informing their neighborhood about the change in the spatial distribution of the ground color. Robots which receive this message perform the same behavior by switching to the adaptive threshold phase and start exploring. After a short time (the time required for this message to reach the edge of the swarm) all robots start moving around, computing their new local thresholds and searching for the new edges to mark. This what we are capturing in Figure 8, that depicts the system at different time steps, including: (i) the robots wandering while computing their local thresholds on the bubbles ground, in Figure 8a, (ii) the detection of edges on the bubble ground, in Figure 8b,c, (iii) the switch to the adaptive threshold phase when the change in the spatial distribution of the ground color was detected. Here, we can notice that robots have started from their positions on the edges, which they have marked in the first environment, see Figure 8d, in which the robots with the red LEDs start from the edges of the marked bubbles. (iv) the computation of the new local thresholds for the flower environment, in Figure 8e, and (v) the detection of edges on the flower ground progressively, in Figure 8f,g,h. On the contrary to abrupt changes in the environment, we have validated our solution in a dynamic environment that changes gradually by enlarging the size of the bubbles in the bubbles ground over three discrete images. The results are depicted in Figure 9 and show a high degree of adaptivity of the robotic system while marking and re-marking the environmental edges.



(a) (b) (c)

Fig. 7: Edge detection using a swarm of 300 robots performed with different values of α (see in Eq. (2)). (a) $\alpha = 0.9$, (b) $\alpha = 0.5$, and (c) $\alpha = 0$. The yellow rectangles surround detected edges which will be missed when lower α(s) are applied, whereas the white rectangles surround the missed edges when higher α(s) were applied.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 8: Several snapshots taken over the simulation time for environmental edge detection with a swarm of 300 robots in a dynamic environment that experienced an abrupt change.

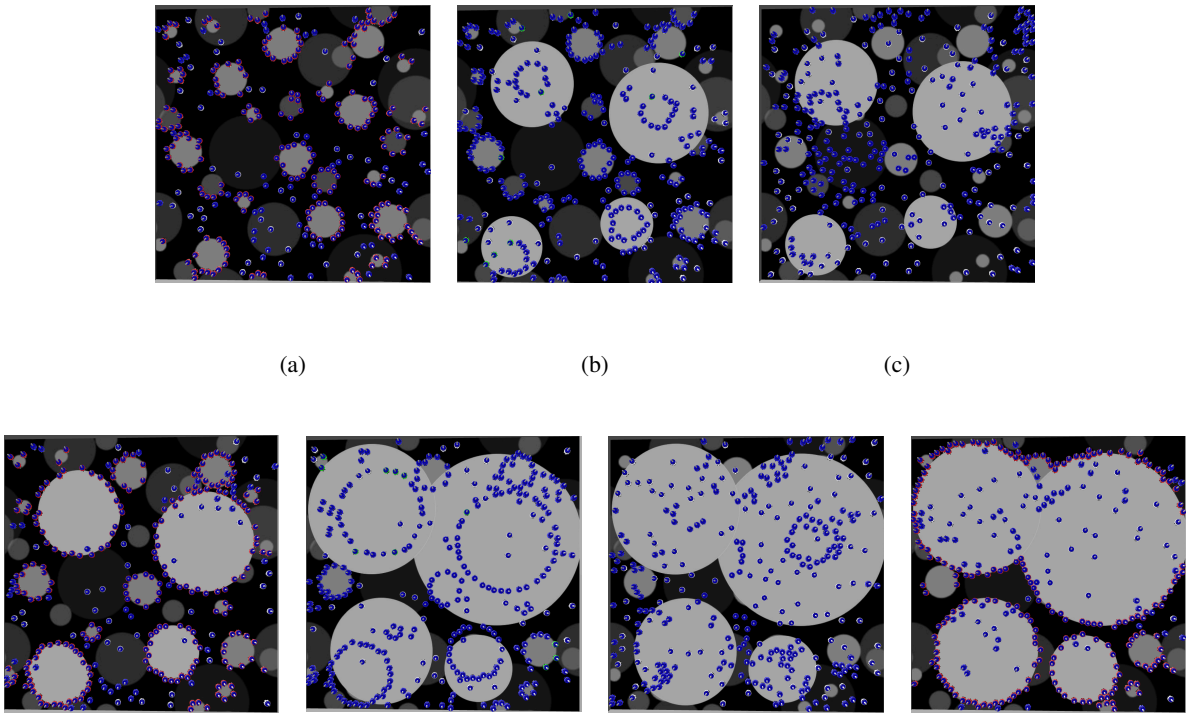
V. Conclusion

In this paper, we have focused on the problem of environmental edge detection using a swarm of homogeneous robots. We have tackled both static and dynamic environments, in which the spatial distribution of the feature can change over time. The highlighted problem can be a fundamental requirement in a large spectrum of applications, in which robots are used to select, isolate, or detect particular parts of an unknown environment. This selection can be required based on a set of specific features, which the robots can perceive distributively. In our paper, we have presented a collective solution that allows robots to, first, explore their environment, second, decide collectively on a threshold that allows them to detect any change in the intensity of the environmental features as a true or a false edge. Finally, to search their environment, and mark the true edges.

Our solution has shown it scalability through being efficient while changing the size of the swarm. It has shown its high level of efficiency in detecting and marking edges with different level of detection based on the system requirements. Furthermore, we have demonstrated the high flexibility of our solution in identifying changes in the spatial distribution of the environmental features in the context of dynamic environments and its ability to adapt to such changes.

Acknowledgements

The authors would like to thank *Syam Gullipalli* of the university of Paderborn, Paderborn, Germany for his help on previous versions of this work.



(a) (b) (c)

(d) (e) (f) (g)

Fig. 9: Several snapshots taken over the simulation time for environmental edge detection with a swarm of 300 robots in a dynamic environment that change gradually.

REFERENCES

[1] G. Beni, "From swarm intelligence to swarm robotics," in *International Workshop on Swarm Robotics*. Springer, 2004, pp. 1–9.

[2] G. Pini, A. Brutschy, M. Birattari, and M. Dorigo, "Interference reduction through task partitioning in a robotic swarm," in *Sixth International Conference on Informatics in Control, Automation and Robotics–ICINCO*, 2009, pp. 52–59.

[3] Y. Khaluf, M. Birattari, and F. Rammig, "Analysis of long-term swarm performance based on short-term experiments," *Soft Computing*, vol. 20, no. 1, pp. 37–48, 2016.

[4] E. Şahin, "Swarm robotics: From sources of inspiration to domains of application," in *International workshop on swarm robotics*. Springer, 2004, pp. 10–20.

[5] M. S. Couceiro, P. A. Vargas, R. P. Rocha, and N. M. Ferreira, "Benchmark of swarm robotics distributed techniques in a search task," *Robotics and Autonomous Systems*, vol. 62, no. 2, pp. 200–213, 2014.

[6] Y. Khaluf and M. Dorigo, "Modeling robot swarms using integrals of birth-death processes," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 11, no. 2, p. 8, 2016.

[7] W. Liu and A. F. Winfield, "Modeling and optimization of adaptive foraging in swarm robotic systems," *The International Journal of Robotics Research*, vol. 29, no. 14, pp. 1743–1760, 2010.

[8] N. Hoff, R. Wood, and R. Nagpal, "Distributed colony-level algorithm switching for robot swarm foraging," in *Distributed Autonomous Robotic Systems*. Springer, 2013, pp. 417–430.

[9] A. Howard, M. J. Matarić, and G. S. Sukhatme, "An incremental self-deployment algorithm for mobile sensor networks," *Autonomous Robots*, vol. 13, no. 2, pp. 113–126, 2002.

[10] H. Hamann and H. Wörn, "A framework of space–time continuous models for algorithm design in swarm robotics," *Swarm Intelligence*, vol. 2, no. 2, pp. 209–239, 2008.

[11] T. Schmickl, R. Thenius, C. Moslinger, J. Timmis, A. Tyrrell, M. Read, J. Hilder, J. Halloy, A. Campo, C. Stefanini *et al.*, "Cocoro–the self-aware underwater swarm," in *Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2011 Fifth IEEE Conference on*. IEEE, 2011, pp. 120–126.

[12] S. Hauert and S. N. Bhatia, "Mechanisms of cooperation in cancer nanomedicine: towards systems nanotechnology," *Trends in biotechnology*, vol. 32, no. 9, pp. 448–455, 2014.

[13] C. Devigne, P. Broly, and J.-L. Deneubourg, "Individual preferences and social interactions determine the aggregation of woodlice," *PLoS One*, vol. 6, no. 2, p. e17389, 2011.

[14] R. Jeanson, C. Rivault, J.-L. Deneubourg, S. Blanco, R. Fournier, C. Jost, and G. Theraulaz, "Self-organized aggregation in cockroaches," *Animal behaviour*, vol. 69, no. 1, pp. 169–180, 2005.

[15] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 207, no. 1167, pp. 187–217, 1980.

[16] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[17] Y. Khaluf and S. Gullipalli, "An efficient ant colony system for edge detection in image processing," in *Advances in Artificial Life, ECAL*, 2015, pp. 398–405.

[18] W. Kerr, D. Spears, W. Spears, and D. Thayer, "Two formal gas models for multi-agent sweeping and obstacle avoidance," in *International Workshop on Formal Approaches to Agent-Based Systems*. Springer, 2004, pp. 111–130.

[19] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, M. Birattari, L. M. Gambardella, and M. Dorigo, "ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems," *Swarm Intelligence*, vol. 6, no. 4, pp. 271–295, 2012.