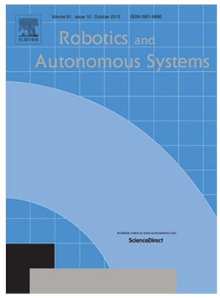


Supervised morphogenesis: Exploiting morphological flexibility of self-assembling multirobot systems through cooperation with aerial robots

Nithin Mathews, Anders Lyhne Christensen, Alessandro Stranieri, Alexander Scheidler, Marco Dorigo

PII: S0921-8890(17)30372-X
DOI: https://doi.org/10.1016/j.robot.2018.11.007
Reference: ROBOT 3112

To appear in: *Robotics and Autonomous Systems*

Received date : 19 June 2017
Revised date : 19 September 2018
Accepted date : 9 November 2018

Supervised morphogenesis: exploiting morphological flexibility of self-assembling multirobot systems through cooperation with aerial robots

Nithin Mathews[1], Anders Lyhne Christensen[2], Alessandro Stranieri[1], Alexander Scheidler[3], and Marco Dorigo[1]

[1]IRIDIA, Université Libre de Bruxelles, 50 Avenue F. Roosevelt, CP 194/6, 1050 Bruxelles, Belgium
{nmathews,mdorigo}@ulb.ac.be

[2]Embodied Systems for Robotics and Learning at the Maersk Mc-Kinney Moller Institute, University of Southern Denmark (SDU), Odense, Denmark; and Instituto Universitário de Lisboa (ISCTE-IUL), Av. das Forças Armadas, 1649-026 Lisboa, Portugal
anders.christensen@iscte.pt

[3]Fraunhofer Institute for Wind Energy and Energy System Technology, Königstor 59, 34119 Kassel, Germany
alexander.scheidler@iee.fraunhofer.de

Abstract

Self-assembling robots have the potential to undergo autonomous morphological adaptation. However, due to the simplicity in their hardware makeup and their limited perspective of the environment, self-assembling robots are often not able to reach their potential and adapt their morphologies to tasks or environments without external cues or prior information. In this paper, we present *supervised morphogenesis* — a control methodology that makes self-assembling robots truly flexible by enabling aerial robots to exploit their elevated position and better view of the environment to initiate and control (hence *supervise*) morphology formation on the ground. We present results of two case studies in which we assess the feasibility of the presented methodology using real robotic hardware. In the case studies, we implemented supervised morphogenesis using two different aerial platforms and up to six self-assembling autonomous robots. We furthermore quantify the benefits attainable for self-assembling robots through cooperation with aerial robots using simulation-based studies. The research presented in this paper is a significant step towards realizing the true potential of self-assembling robots by enabling autonomous morphological adaptation to a priori unknown tasks and environments.

Keywords — Self-assembling robots, heterogeneous multirobot teams, distributed systems, air/ground robot teams, robot coordination, modular robots

1 Introduction

Self-assembling robots have the physical features [1, 2, 3, 4] and control algorithms [5, 6, 7, 8] that enable them to form distinctive collective robot structures (hereafter referred to as *morphologies*) by physically connecting to one another. This morphological flexibility inherent to self-assembling multirobot systems is the prime motivation why researchers are interested in studying such systems. However, existing self-assembling robots are often pre-programmed by human operators who precisely define the scale and shape of morphologies to be formed (hereafter referred to as *target morphologies*) prior to deployment [9] or rely on specific environmental cues for the robots to infer target morphologies [10]. This is primarily because self-assembling robots tend to be relatively simple robotic units that lack the sensory apparatus to characterize the environment with sufficient accuracy to autonomously find a suitable target morphology for a given situation. Existing self-assembling robots therefore remain limited in exploiting their morphological flexibility and fail to realize their full potential.

To overcome these limitations, we propose *supervised morphogenesis* – a control methodology in which we extend the functionality of a group of self-assembling robots by an aerial robot to which we delegate decision-making authority. That is, self-assembling robots rely on aerial robots to act as an "eye-in-the-sky" and to provide the guidance required to form new morphologies as a function of the task and/or the environment. Many researchers have considered such air-ground teams [11, 12] as they have the potential to solve tasks that require capabilities that go beyond those of a single robot type. For instance, while aerial robots can explore large areas rapidly, ground-based robots can carry higher payloads and manipulate objects on the ground. In recent years, a surge in technology has led to the development of aerial robots [13] able to maneuver in previously unreachable environments such as inside buildings including obstacle-filled factory halls and warehouses [14, 15, 16]. Innovative designs have also been proposed [17, 18] rendering aerial robots resilient towards potential collisions in cluttered environments.

As pointed out by Lacroix and Besnerais [12], the deployment of air-ground robot teams calls for the resolution of multiple challenges related to perception, decision, and action. By definition, members of such teams operate in different spaces with varying vantage points of the environment. Reaching team-level decisions based on the environment perceived in this manner poses a crucial challenge to air-ground teams. This is particularly true when such teams operate in environments in which they are not able to share a common frame of reference (such as GPS) or when they execute missions that require higher precision than that offered by civilian GPS systems. Supervised morphogenesis proposes a solution to each of the three challenges pointed out by Lacroix and Besnerais so that self-assembling robots can operate in a priori unknown environments and rely on an aerial robot for ad-hoc assistance. The supervised morphogenesis control methodology that we propose in this paper allows decentralized air-ground robot teams to cooperate without relying on external infrastructure or GPS. Potential application scenarios include collaborative monitoring and surveillance in large warehouses or outdoor environments, search-and-rescue missions, and autonomous exploration and mapping of unknown terrains.

In supervised morphogenesis, aerial robots exploit their elevated position to characterize the environment and its features so that they can supervise the formation of suitable target morphologies. In our study, aerial robots use standard monocular cameras to observe the environment. A two- or three-dimensional model of the environment is generated from these observations depending on the task. Subsequently, the environment model is used to perform *on-board simulations* to determine if and when self-assembly is required. The simulations are also used to determine the shape and size of target morphologies. These simulations allow aerial robots to assess the performance of different candidate morphologies in a particular environment prior to their costly (in terms of energy and time) realization on the ground. Spatially targeted communication [19] is then applied to let aerial robots establish a communication link to specific robots based on their location on the ground. Morphology formation instructions [7] are then transmitted through this link to initiate the formation of target [20]. We present two case studies in which we assess the feasibility of the proposed control methodology using real robotic hardware. Furthermore, we quantify the benefits of cooperation between an aerial robot and self-assembling robots using simulation-based experiments [21]. We show that the presented control methodology allows self-assembling robots to adapt to previously unknown tasks and environments by cooperating with an aerial robot.

2 Related Work

In this section, we review robotic systems that use technologies similar to the ones presented in this paper with a focus on air-ground teams. We also present control frameworks developed particularly for the control and coordination of teams composed of aerial and ground-based robots. However, to the best of our knowledge, cooperation between aerial robots and self-assembling robots has not been previously studied.

O'Grady et al. [22] presented a robotic system composed of homogeneous self-assembling robots able to cross a gap. In the study, however, the response behavior of the robots when encountering a gap was pre-programmed. The robots did not possess the sensory apparatus required to estimate the gap width. Therefore, on encountering a gap, they self-assembled into a chain morphology of a pre-programmed size irrespective of the width of the gap, and hence, decreased the efficiency of the whole system with respect to task completion times. A more effective self-assembling robot system was presented in Mathews et al. [23]. The authors proposed a novel paradigm for cooperation that allows robot controllers to be merged into a single control structure within a heterogeneous group of robots. They presented a form of cooperation that has the potential to merge sensory or guidance information from aerial robots into the control of robots operating on the ground. Application scenarios for such air-ground robot systems have already been presented in [20].

Aerial robots equipped with monocular vision cameras have been used to compute height maps of ground surfaces [24, 25]. Lacroix et al. [24] presented a pioneering work in which a tethered blimp is flown at an altitude between 10 and 40 m to retrieve stereo images used to compute the height map of an area covering several thousands of square meters. Forster et al. [25] showed how an aerial robot can use two different monocular vision streams to compute height maps at 1 Hz. The robot used the height maps to detect safe landing-spots and to autonomously land. One of the few examples of aerial robots able to navigate through indoor environments and equipped with a Kinect sensor was presented in [26]. This work considered a scenario in which an air-ground robot team is used to map a damaged building from the inside. Note that the height maps computed by aerial robots in [24, 25, 26] were not included in the decision-making processes of robots operating on the ground.

Kim et al. [27] showed how two aerial robots can provide the stereo vision to a peer robot on the ground that then can compute height maps and use them in the robots decision-making processes. However, contrary to a decentralized decision-making mechanism [28], the approach presented by Kim et al. may not be scalable for systems that consider robots that operate in groups of rather large sizes. In such a system, vision streams would have to be transmitted on a per robot basis and may cause bandwidth issues as group-size increases.

A search-and-rescue task was solved by two ground-robots cooperating with an aerial robot in [29]. Dorigo et al. [30] proposed a heterogeneous robot team that includes a climbing robot besides aerial and ground-based robots. This additional robot type was able to climb along indoor vertical surfaces and manipulate objects unreachable by the two other robot types. A search-and-retrieve experiment is presented in which over twenty robots are able to combine their different capabilities to locate and retrieve a book situated on a book shelf. Langerwisch et al. [31] presented a heterogeneous team composed of a car-sized robot and two quadcopter drones. Through a centralized control station that maintains communication contact to all three vehicles at all times, a human operator was able to issue surveillance tasks at the team level. The system was demonstrated in outdoor environments and requires GPS.

Contrary to the system presented in this paper, air-ground teams are often composed of a single terrestrial and a single aerial robot. Such systems impose lower requirements on coordination and communication mechanisms due to the limited number of team members. Nevertheless, they have been successfully applied to solve various tasks such as to cooperatively map obstacles in large areas [32], to augment the view of a moving ground-based robot with aerial images [33, 34], and to cooperatively track a moving target [35]. In more recent work, Käslin et al. [36] presented a localization method based on elevation maps for ground robots. The method is independent of sensors and allows a ground robot to find its relative position and orientation within the reference map provided by an aerial robot without relying on GPS.

A customizable framework to enable collaboration between aerial and terrestrial drones was presented in [37]. The framework was validated in a real-world search-and-find scenario in which team members detected each others presence, selected leaders of a team, and assigned tasks to particular members in the team. Saska et al. [38] proposed a control scheme that allows an air-ground team to coordinate and control its members in a leader-follower scenario. The scheme enables the followers of the leader to also maintain a particular formation throughout the whole mission. The scheme was validated using numerous search-and-rescue scenarios both in simulation and in the real world. Although with a human operator in the loop, Hsrik et al. [39] proposed a decentralized architecture that enables interaction between an aerial robot providing global coverage and a couple of ground-based robots providing local coverage of a monitored environment. The architecture was validated in an area-inspection scenario. Although these control frameworks and schemes have been validated using different application scenarios, they cannot be immediately applied to air-ground teams that cooperate to form adaptive morphologies on the ground through self-assembly.

3 The robot platforms

We use three different robot platforms in this study: two aerial robots and one self-assembling robot. We summarize the main specifications of the robot types in Table 1 and describe them in more detail in the following.

The AR.Drone [40] is a quadcopter (see Fig. 1a) with a front-facing camera and a downward-pointing camera (176x144 @ 60fps). The AR.Drone has an autonomy of up to 12 minutes while flying at speeds of up to 18 km/h. The robot's processing unit is an ARM9 running at 468 MHz with 128 MB of DDR RAM. Other features include a six degrees of freedom inertial measurement unit and an ultrasound altimeter. An API provides access to sensory information including altitude and battery level from the AR.Drone. External devices such as a PC can therefore retrieve these information and simultaneously communicate with the AR.Drone at 30 Hz via an ad-hoc wireless Ethernet network. Note that also images from both cameras can be retrieved from the AR.Drone at the same fre-

	AR.Drone	eye-bot	foot-bot
Dimension	57 cm across, ht. 12 cm	50 cm ⌀, ht. 54 cm	17 cm ⌀, ht. 17 cm
Weight	380 g	max. 2 kg (incl. payload)	ca. 1 kg
Processor	ARM 9 (468 MHz)	ARM11 (533 MHz)	ARM 11 (533 MHz)
RAM	128 MB	128 MB	128 MB
Vision	176x144 / 640x480	2 MP 360° pan-and-tilt	3 MP / 3 MP
Autonomy	ca. 12 min	ca. 20 min	4 h to 7 h

Table 1: Hardware specifications of the robot platforms used in this study.

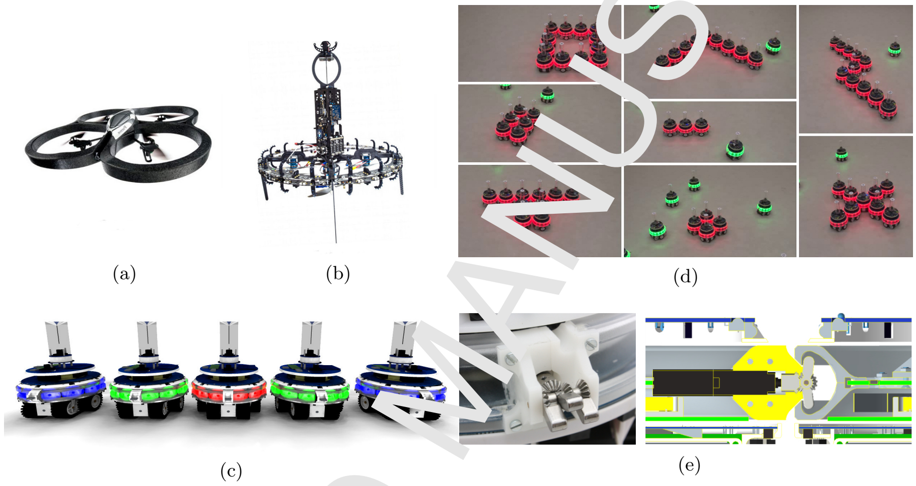


Figure 1: The robot platforms used in this work. (a) The AR.Drone. (b) The eye-bot. (c) Five foot-bots capable of self-assembly with their LEDs illuminated in different RGB colors. (d) A photo montage showing example morphologies that can be formed by multiple foot-bots. (e) The three-fingered connection device of the foot-bot before and after insertion into the passive docking ring of another robot.

quency. We used the software development kit presented in [41] to channel video streams from the AR.Drone to a remote PC where vision algorithms were executed. Position control data computed on the basis of these streams were then transmitted back to the AR.Drone in real time via wireless Ethernet.

Figure 1b shows the eye-bot [42] aerial robot. Its thrust and control are provided by eight rotors mounted in a co-axial quadrotor configuration. The carbon fiber structure of the eye-bot weighs only 270 g and is able to lift a payload of up to 2000 g – sufficient for the mounting of a range of advanced sensors. The on-board battery provides the eye-bot with up to 20 minutes of autonomy depending on payload. The eye-bot's most unique feature, however, is a ceiling attachment system based on active magnets that allows the eye-bot to extend its autonomy considerably [42] by attaching to metal ceilings or bars. Other features include a downward-pointing 2 MP 360° pan-and-tilt camera that allows the eye-bot to survey the ground underneath it for other robots and objects, a ring encompassing the robot's chassis with 16 built-in RGB LEDs, an altitude sensor, and a magnetometer to detect its own orientation. The eye-bot is also equipped with a 3D relative positioning and communication device [43]. This on-board device has a maximum range of 12 m and allows an eye-bot to communicate with other eye-bots in flight and to detect walls and other obstacles.

The foot-bot (see Fig. 1c) is a particular configuration of the ground-based marXbot platform [4]. The marXbot platform (diameter 17 cm) consists of a series of sensor and actuator modules that can be combined into particular robot configurations depending on task requirements. In the foot-bot configuration, the robot is equipped with an ARM 11 processor (i.MX31 clocked at 533 MHz and with 128 MB RAM) running a Linux-based operating system, 12 RGB-colored LEDs, a 2D distance scanner, 24 IR proximity sensors, a 3-axis gyroscope, one omni-directional (3 MP) and one ceiling (3 MP) camera. The self-assembly module includes a rotatable docking component composed of an active docking unit with three fingers and a passive docking ring. A physical connection is formed when a foot-bot inserts its docking unit into the docking ring of another foot-bot and then opens its three fingers. Figure 1e shows examples of different morphologies foot-bots can form. A key novelty of the foot-bot is its range and bearing communication device [44] (referred to as mxRAB device in the following). The mxRAB device allows the simultaneous estimation of relative positions (i.e., the range and bearing) of peer robots.

The foot-bots use the mxRAB device for communication with each other. The device enables situated communication at 10 Hz and can estimate the range and bearing of message sending robots at a distance of up to 5 m. For communication between the AR.Drone and the foot-bots, we rely on standard wireless Ethernet broadcast. As the number of Ethernet devices able to connect directly to the AR.Drone network is limited to one, we route the messages through a PC connected to both the AR.Drone and the foot-bots network. The eye-bot, on the other hand, is able to communicate directly through broadcast wireless Ethernet messages to the foot-bots. If no wireless Ethernet is available, the eye-bot can transmit messages in the form of colors displayed on its 16 LEDs. In this case, the foot-bots use their upward-pointing cameras to detect the messages sent by the eye-bot at intervals of 300 ms. Similarly, the foot-bots display different colors using their LED rings to send signals to the aerial robots. To detect these signals, we retrieve the video stream from the AR.Drone's downward-pointing camera using a PC and run off-board vision algorithms at 16 Hz (i.e., new signals can be detected approximately every 60 ms). The eye-bot, on the other hand, requires up to 3 s to process each image captured using its 360° HD pan-and-tilt camera.

4 Control methodology

We developed one controller for each robot type (i.e., aerial and ground-based robots) used in this study to enable supervised morphogenesis. As shown in Fig. 2, the control of each robot type transitions through multiple control states (drawn in circles). In previous research, we developed several of these control states with the goal of facilitating supervised morphogenesis. In the rest of this section, we describe each control state. In the following Sections 5 and 6, we present two case studies that demonstrate the control methodology presented in this section using real robot hardware.

4.1 Build environment model

An aerial robot hovers above the group of foot-bots it supervises and builds an internal model of the environment in its field of view. The dimensionality of the model, i.e., 2D or 3D, depends on the considered task. For instance, certain tasks (see Sect. 5) require the

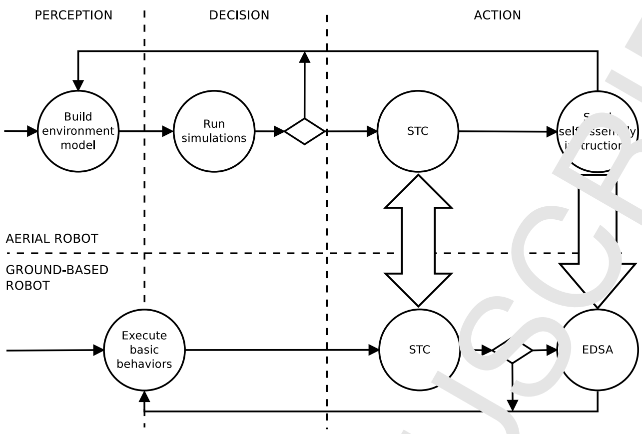


Figure 2: An overview of supervised morphogenesis. Each circle represents a control state related to perception, decision or action taken by the heterogeneous team. Basic behaviors (such as phototaxis and obstacle avoidance) depend on the actual task and may include perception and decision-related activities that do not require aerial supervision. The vertical arrows indicate the occurrence and directionality of interactions between the robot types. STC: spatially targeted communication, EDSA: enhanced directional self-assembly.

aerial robot to model only the relative positions of the foot-bots and other objects in the environment, while other tasks (see Sect. 6) require a detailed three-dimensional model of the environment. We use standard monocular cameras, which are available on most aerial platforms, to build the models.

4.2 Run simulations

When a model of the environment has been built, the aerial robot runs simulations to determine whether or not the robots on the ground require supervision to solve the task at hand or to maneuver through an environment. The on-board simulations also allow the aerial robot to evaluate the adequacy of candidate morphologies prior to their physical formation on the ground, which is costly in terms of time and energy. If the outcome of the simulations requires no action to be taken, the aerial robot continues with the modeling of the environment.

4.3 Spatially targeted communication (STC)

The aerial robot needs to be able to communicate with the robots on the ground to supervise formation of an appropriate morphology. Ideally, the communication is targeted to a particular set of robots such that (i) robots that should self-assemble can be directly addressed, and (ii) resources are not allocated unnecessarily allowing robots not required for self-assembly to pursue other tasks. For this purpose, we developed spatially targeted communication (STC) [19]. STC allows a robot in a multirobot system deprived of GPS and global maps to establish a dedicated communication link to another robot based on location. Using messages exchanged via LEDs and cameras, such links can be established between a ground-based robot group [45], as well as in an heterogeneous group composed of both aerial and ground-based robots [19]. At the core of establishing an STC link lies an iterative elimination process. An iterative growth process can be then executed to add further co-located robots to an existing STC link. In supervised morphogenesis, we let the aerial robot establish an STC link to the foot-bot best located to initiate the self-assembly process of the target morphology. Ground-based robots with which no links were established resume their individual task-related behavior.

4.4 Send self-assembly instructions

The aerial robot uses an established STC link to broadcast self-assembly instructions that lead to the formation of a target morphology. The instructions are described using an improved version of SWARMORPH-script [7] — a descriptive language that is executed by autonomous self-assembling robots and can describe arbitrary morphologies. SWARMORPH-scripts can also be compiled into a morphology library that can be preloaded on the foot-bots. Depending on the task, an aerial robot can then activate a particular morphology over an STC link by transmitting a single message that then can be mapped to a target morphology using a lookup table available to both communicating robots. After successful transmission, the control of the aerial robot returns to the component responsible for modeling the environment.

SWARMORPH-script was initially developed for a self-assembling robotic platform [46] that preceded the foot-bots and was limited to LEDs and camera-based communication between robots. In this study, we extended the technology behind SWARMORPH-script to take advantage of the higher communication bandwidth and speed provided by the mxRAB device available to the foot-bots. These enhancements allow foot-bots to demonstrate behaviors that were unachievable to their predecessors such as forming multiple connections in parallel [20] and coordinating the motion in target morphologies [20, 47, 48].

4.5 Enhanced directional self-assembly (EDSA)

Larger morphologies can only be formed by self-assembling robots if connections can be formed between connection-inviting robots and connection-seeking robots. We developed enhanced directional self-assembly (EDSA) [20] as a connection forming mechanism for the foot-bots. The mechanism takes advantage of the high-speed situated communication provided by the mxRAB device. It is based on a recruitment and guidance-based algorithm that enables the foot-bot initiating the self-assembly process to invite suitably located neighboring robots to form direction specific connections at angles described in the SWARMORPH-script being executed.

4.6 Execute basic behaviors

We refer to behaviors that do not require supervision from the aerial robot as *basic behaviors*. Based on data acquired through its sensors, robot-level decisions are made and then translated into actuator commands by a robot executing basic behaviors. Examples of basic behaviors include obstacle avoidance and phototaxis.

5 Case study 1: supervision based on a 2D environment model

The goal of the first case study is to validate that a physical aerial robot can supervise foot-bots according to an a priori unknown configuration of the environment, based on the

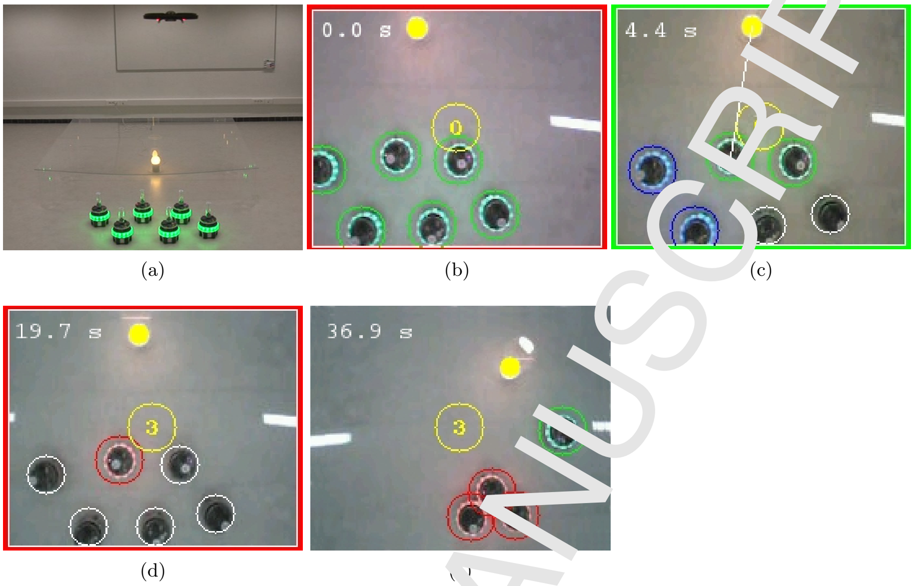


Figure 3: The experimental setup of case study 1. (a) The robot team is composed of one AR.Drone and six foot-bots. A light source is placed on the ground. Figures (b-d) are frames taken from the AR.Drone camera when executing STC. Border colors visualize transmitted messages; the number in the center of each frame shows the size of the target morphology resulting from the simulations run by the AR.Drone. Figure (e) shows the target morphology formed using EDSA. The times shown in Figures (b-e) correspond to the clock time starting the execution of STC. For safety reasons, a transparent plexiglass platform is installed at 40 cm height in the area of the light source in order to shield the foot-bots from the AR.Drone executing emergency landing behaviors.

vision and communication system developed in this study. For this purpose, we designed a task in which an AR.Drone should first locate a light source in the environment and then estimate the total number of foot-bots. The AR.Drone should then instruct a subset of the foot-bots to construct a morphology at a certain distance (between 60 cm and 70 cm) from the light source. Since the AR.Drone has no a priori information about the configuration of the environment or the number of foot-bots, the task assesses the aerial robot's capacity to correctly detect and estimate relative distances between objects in the environment, and to supervise self-assembling robots on the ground.

In the following, we describe in detail a successful experimental run using the snapshots presented in Fig 3. The foot-bots start with their LEDs illuminated in green. The intense ambient lighting in the environment does not permit the foot-bots to detect the light source nor can they detect neighboring robots using their cameras. The foot-bots are initially placed facing the light source and instructed to move forward.

The AR.Drone flies ahead of the foot-bots and scans the environment for the light source. Once it has detected the light source, the AR.Drone waits for the foot-bots to arrive by hovering above the light source. When the first foot-bot enters its field of view, the AR.Drone starts running simulations. If the relative distance between any of the foot-bots and the light source is between 60 cm and 70 cm, the AR.Drone broadcasts a *stop* command in the form of a SWARMORPH-script. All foot-bots receive and execute this command and come to a halt (see Fig. 3b). Subsequently, the AR.Drone scans the area around the already detected foot-bots to estimate the total number of foot-bots in the group from which the size of the target morphology — i.e., the number of robots that need to attach to each other — is computed. We designed the task such that the aerial robot should leave three robots unconnected and ready for other tasks. The AR.Drone then initiates the protocol described in Sect. 4.3 to select the closest foot-bot (indicated using a straight line in Fig. 3c) to the light source. Once the communication link is established (see Fig. 3d), a SWARMORPH-script containing the instructions to build a target morphology of size three is chosen from a preloaded morphology library and then transmitted to the selected foot-bot. The foot-bot executes the SWARMORPH-script it received to form a triangle morphology of size three (see Fig. 3e).

We carried out 10 experimental runs using one AR.Drone and six foot-bots. In seven of the 10 runs, the aerial robot successfully completed the task. In the remaining three runs, the AR.Drone did not detect all the foot-bots present in the environment. Note that these vision-related issues occurred because of dimmer foot-bot LEDs caused by low battery voltage on some of the foot-bots. The occurrence of such faulty runs could be reduced in future experiments by only deploying foot-bots with a certain battery voltage level or by replacing the low-resolution vertical AR.Drone camera (176x144) with a high-resolution camera supporting the detection of foot-bots with dimmer LEDs.

Note that the AR.Drone chose a suitable SWARMORPH-script describing the target morphology from a preloaded library containing multiple morphologies of different shapes and sizes. In Sect. 6, we show that task-dependent morphologies can be determined and generated by aerial robots on-the-fly based on observed environmental features. Also, note that the AR.Drone was flown manually while all other control components and the foot-bots were entirely autonomous. Video footage of this experiment can be found online [49]. Further examples of cooperation between an AR.Drone and foot-bots were presented in [48].

6 Case study 2: supervision based on a 3D environment model

The goal of the second case study is to validate supervised morphogenesis in a scenario in which, to be successful, the heterogeneous robot team needs to consider also physical characteristics of the environment. For this purpose, we consider a more challenging hill-crossing task in which we deploy five foot-bots in an environment containing an elevated surface, hereafter referred to as *hill-obstacle*. For the team to be successful, the aerial robot is required to build a three-dimensional model of the environment to detect and characterize the obstacle. The three-dimensional model of the environment requires the aerial robot to run more sophisticated simulations while also considering the limited computational capabilities available on-board. To test the feasibility of supervised morphogenesis in heterogeneous teams that are limited to a low-bandwidth communication modality, we further limit both robot types to LEDs and cameras-based communication. The task also requires the formation of multiple target morphologies — rather than a single one as in the previous case study. For the experiments carried out in this case study, we use one eye-bot and a group composed of five foot-bots.

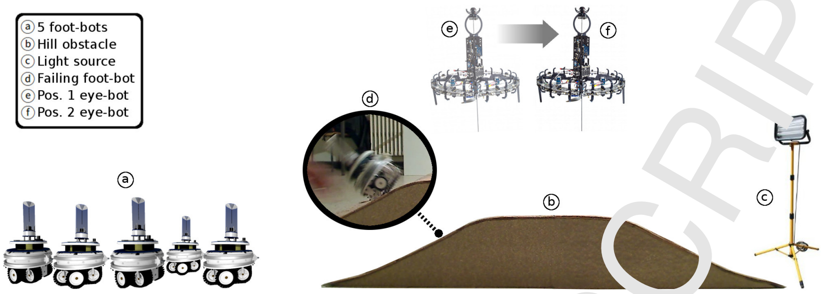


Figure 4: The experimental setup of the hill-crossing task. Five foot-bots are shown in the deployment area. A light source representing the destination area is shown on the right. A hill obstacle that cannot be crossed by an individual foot-bot is shown between the areas. Visualized are also two positions above the hill obstacle the eye-bot considers when using its monocular vision system.

6.1 Experimental setup

We consider a hill-crossing task that requires the foot-bots to move towards a light source from a deployment area where they are initially placed (see Fig. 4a–c). They use their light sensors to detect the light source and move towards it. As shown in Fig. 4d, we place a steep hill in their path. The hill may be so steep that a foot-bot topples over if it tries to cross the hill alone. We vary the hill steepness between 0° (i.e., the obstacle is absent) and 30°. Individual foot-bots can withstand a maximum inclination of 25° without toppling over. If the inclination exceeds 25°, the foot-bots have to self-assemble into morphologies that can provide the physical stability required to cross the hill obstacle. As the foot-bots can neither detect the presence of the hill obstacle nor know the size of the group (required to determine the shape and the size of target morphologies that need to be formed), they depend on the eye-bot to provide the guidance necessary to reach the light source. The task is considered to be solved if all five foot-bots manage to reach the light source without toppling over. We assume that the eye-bot has flown in advance and attached to the ceiling over the hill obstacle before the foot-bots reach the obstacle.

6.2 Modeling the environment using height maps

On its flight ahead of the foot-bots to the light source, the eye-bot continuously builds and updates a three-dimensional model of the surface underneath it by computing a height map. We consider two different methods: (i) compute height maps using stereo images acquired through a standard monocular camera, and (ii), extract height maps from a dedicated sensor – the Microsoft Kinect. The extraction of three-dimensional information based on stereo images has been thoroughly studied by the computer vision community [50]. The Kinect is a commercially available RGB-D camera. More details on the two methods and a quantitative comparison between the two are provided in the Appendix.

6.3 On-board simulation-based decision-making

From each newly computed height map, the eye-bot first constructs a height profile by reading cell values along each foot-bot estimated trajectory. The estimated trajectory is assumed to be a straight line connecting a foot-bot's current position to the light source

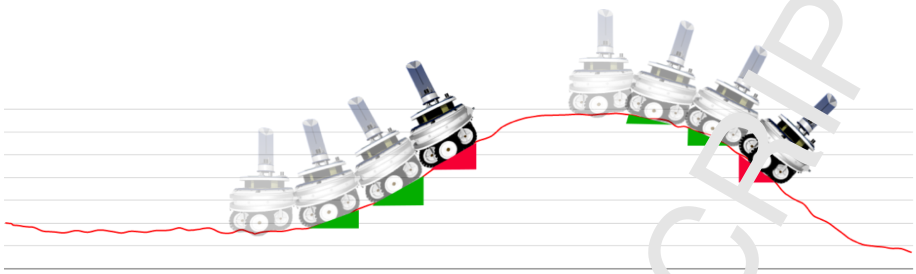


Figure 5: Visualization of a simulation run executed by the eye-bot. A foot-bot is shown at different positions along its estimated trajectory. The simulation is shown to detect at least two areas with a too steep inclinations (marked in red) rendering the whole trajectory too dangerous for the foot-bot.

	no hill	gentle slope	steep hill
number of trials	10	10	10
number of successful runs	10	10	7

Table 2: Summary of the experiment results obtained in the second case study.

in the eye-bot's field of view. Then, the eye-bot simulates a passage of each foot-bot by moving it pixel-by-pixel along the height profile while also computing the inclination each time the foot-bot is moved. These simulations enable the eye-bot to estimate the stability of a foot-bot on the ground on its way towards the light source. An example is visualized in Fig. 5. The foot-bot is first placed at its currently detected position on the height profile and the inclination experienced by the foot-bot at this particular segment is calculated. The simulation ends when the foot-bot's chassis reaches the light source or when a calculated inclination for a foot-bot exceeds 25 °, the maximum inclination angle an individual foot-bot can endure without toppling over. If an inclination of more than 25 ° is found, the eye-bot takes the necessary actions to bring the foot-bots to halt and instructs them to self-assemble into morphologies that guarantee safe passage over the hill obstacle.

6.4 Experiments and results

We conducted experiments using three different scenarios: (i) no hill obstacle, (ii) one hill obstacle with a gentle slope safe for individual foot-bots to cross, and (iii) one steep hill obstacle not crossable by individual foot-bots. As listed in Table 2, we executed 10 experimental runs for each scenario resulting in a total of 30 experimental runs. In scenarios (i) and (ii) the eye-bot classified the surface to be safe for the foot-bots and did not intervene in any of the 20 runs. As a result, the foot-bots reached the destination in all runs. In scenario (iii), where self-assembly is required to solve the task successfully, the foot-bots only reached their destination in 7 out of 10 runs. Two runs failed because of a broken physical connection (i.e., the docking mechanism) between two neighboring foot-bots in an already formed target morphology due to too high torque causing all three foot-bots to topple over. In a further run, a foot-bot stopped functioning due to low battery charge. These failed runs thus did not result from flaws in the control methodology but were caused by hardware-related issues.

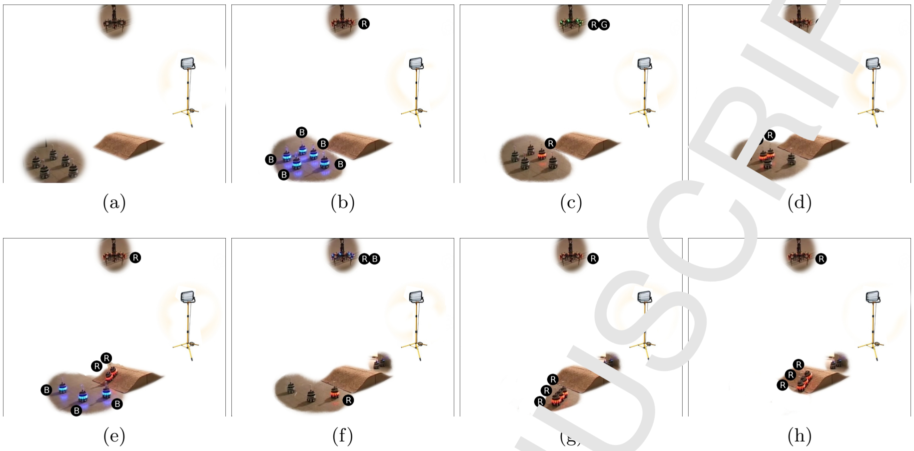


Figure 6: Snapshots of the hill-crossing experiment. For better visibility, background clutter has been edited out and the light source has been visualized schematically. The signals transmitted are indicated as follows: R=RED, G=GREEN, B=BLUE, RG=RED-GREEN, and RB=RED-BLUE. (a) Deployment phase. Foot-bots execute phototaxis behavior. The eye-bot is attached to the ceiling above the hill obstacle running on-board simulations. (b) As the foot-bots reach the hill obstacle, the eye-bot issues the RED signal: the foot-bots halt. Both robot types start executing STC. (c) The communication link is established to a single foot-bot and followed by transmitting the RED-GREEN signal. (d) As a result, the foot-bot executes a SWARMORPH-script that leads to the formation of a chain morphology composed of two foot-bots. (e) As the chain morphology successfully moves over the hill-obstacle, the eye-bots establishes a further communication link to one of the remaining foot-bots and (f) activates the execution of another SWARMORPH-script by sending the RED-BLUE signal. (g) A chain morphology composed of three foot-bots is formed. (h) The task is successfully solved by the team as all foot-bots manage to successfully cross the hill-obstacle.

In the following, we present details of a successful experimental run from scenario (iii). Snapshots from the experiment are presented in Fig. 6. We first manually move the eye-bot between two positions 30 cm away from each other above the hill-obstacle. The eye-bot then computes 10 height maps using the stereo images retrieved from both positions. The simulation ends when all height maps have been considered or, as in this experiment, if the eye-bot's belief β of an hazardous environment is greater than 90%. After the simulated run of each foot-bot in the eye-bot's field of view, β is updated using a simple filtering method: $\beta = (1-c)\cdot\beta_{h-1}+c\cdot\beta_h$, where β_h is a binary value representing the outcome of the simulation (where 0=no danger and 1=danger) and c, $0 \leq c \leq 1$, is the confidence level of the eye-bot in the precision of the underlying height map. We empirically determined $c = 0.85$ to be appropriate for height maps computed using stereo images and $c = 0.9$ for height maps returned by the Kinect sensor. This filtering method makes simulations less vulnerable to extreme outliers and smooths the modeled ground surface. For this particular run, the average value for the maximum inclination computed was [illegible]° with a standard deviation of 2.88°. In the experiment presented here, the hazardous environment was detected on the basis of ten simulated runs in total (i.e., two runs for each foot-bot). Note, that even though in this particular case study the eye-bot control does not require such a high level of precision for decision-making, data precision available to the aerial robots may be crucial in other application scenarios.

After moving the eye-bot, the foot-bot's are deployed. They execute a basic phototaxis behavior. The foot-bots are neither aware of the hazardous situation ahead of them nor are they aware of the total number of foot-bots present in the environment. Following the simulations, the eye-bot establishes an STC link to the foot-bot closest to the hill obstacle (see Fig. 6c). Establishing the STC link took 16 s. This link was used by the eye-bot to send target morphology related information. Given that the total number of foot-bots within the eye-bot's view is five, the eye-bot first sends the RED-BLUE signal to initiate the formation of a chain morphology of size two. This target morphology was formed within 6 s (see Fig. 6d). Once completed, the target morphology executes a collective phototaxis behavior that gets them over the hill obstacle within the next 12 s. In the meantime, the eye-bot establishes another STC link to a second foot-bot (see Fig. 6d and f) and issues a RED-GREEN signal to invoke the formation of a chain morphology of size three. The formation of this target morphology took 15 s (see Fig. 6g) and its successful crossing took another 11 s (see Fig. 6h). The total duration of the experiment was 70 s. Note that in the experiments presented here, the foot-bots were preloaded with a library composed of multiple SWARMORPH-scripts describing a variety of morphologies of different sizes. The target morphologies were autonomously chosen by the eye-bot. Video footage of the experiment can be found online [49].

7 Quantifying performance benefits

In this section, we study the performance benefits, measured in terms of task completion time, of supervised morphogenesis over two other methodologies that either require no supervision from aerial robots or do not consider location-based selection of robots on the ground. For the comparison, we chose a task that can be solved by foot-bots with or without the supervision of an aerial robot. In order to collect a sufficient amount of data for the analysis of the performances of the different control methodologies that we compare, we executed more than 1 000 experimental runs using the simulation framework for heterogeneous robot teams described in [51].

In the remainder of this section, we first present the task, the three control methodologies we used to assess the performance benefits, and finally we present and discuss the results.

7.1 Task and experimental setup

We deploy a robot team composed of an eye-bot and 10 foot-bots in an environment consisting of a start zone, a target zone, and a gap that separates the two zones (see Fig. 7). A light source is located in the target zone. At the start of each experiment, 10 foot-bots are placed at random positions with random orientations within a square area of 2 m x 2 m — the *start zone*. When an experiment begins, the foot-bots use their light sensors to detect the light source (i.e., they execute a phototaxis behavior) and move towards it. Simultaneously, they use their ground sensors to detect the gap. The eye-bot hovers above the group and uses its pan-and-tilt camera to detect potential gaps and to estimate their widths. Depending on the width, the foot-bots may need to physically connect to each other to form a chain morphology to cross the gap. The length in foot-bots of such a chain morphology must be chosen so that it guarantees a safe crossing of the

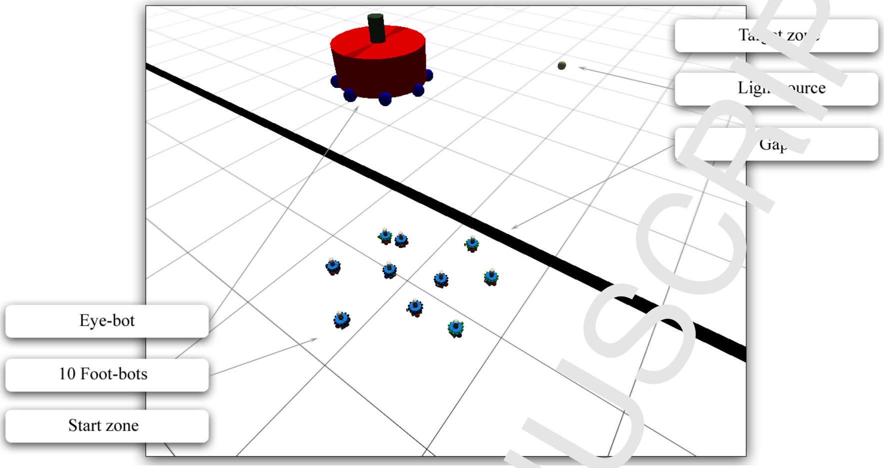


Figure 7: The simulated environment in which the robots operate. The gap separates the environment into a start zone and a target zone with a light source. An eye-bot and ten foot-bots are visible in the start zone.

gap and depends therefore on the width of the gap. In this study, we use gaps of widths of 5 cm, 10 cm, 15 cm and 25 cm. In case of a 5 cm wide gap, individual foot-bots are able to move over it without falling into the gap. For all other widths, the foot-bots are required to form a chain morphology of 2, 3 and 4 foot-bots, respectively, to be successful. The task is considered to be completed when the final foot-bot of a target morphology reaches the target zone.

7.2 Control strategies

In order to quantify the performance benefits of supervised morphogenesis, we have developed three control methodologies. In the first of them — non-cooperative control or NCC — the foot-bots do not seek supervision from the eye-bot to solve the task. In the other two methodologies — namely in location-based supervised morphogenesis (LSM) and supervision based on random groups (SRG) — the foot-bots cooperate with the eye-bot by relying on LEDs and camera-based low-bandwidth communication modality.

Non-cooperative control (NCC): This control methodology is the implementation of the work presented in [22]. The foot-bots are pre-loaded with a SWARMORPH-script that they use to form a four foot-bot chain morphology when a gap (regardless of its actual width) is encountered. The foot-bots do not cooperate nor do they seek for supervision from the eye-bot. They initially move towards the light source until one of the foot-bots detects the gap using its ground sensors. The foot-bot warns neighboring foot-bots via messages sent via the mxRAB device and retreats approximately 40 cm from the gap. Subsequently, it invites neighboring foot-bots to connect to its rear. The other foot-bots stop executing the phototaxis behavior and volunteer to join the ongoing self-assembly process. Once the chain of four foot-bots is formed, the morphology moves towards the light source to cross the gap.

Location-based supervised morphogenesis (LSM): This is the implementation of the control methodology presented in this paper. The foot-bots do not have a priori knowledge about the environment or the target morphology to be formed. They are programmed to execute a phototaxis behavior until messages from the eye-bot are received. In case a gap wider than 5 cm is detected, the eye-bot starts to transmit messages to establish an STC link to a foot-bot that is approximately 40 cm away from the gap. All foot-bots remain static as long as messages are received from the eye-bot. Then, the STC link is extended by the eye-bot to include the foot-bot's neighbors [19] required to form the target morphology. The number of neighbors depends on the gap width. These foot-bots receive a SWARMORPH-script from the eye-bot and follow the instructions in the script to self-assemble into a target morphology the size of which depends on the width of the detected gap. Once the target morphology is formed, the foot-bots move towards the light to cross the gap.

Supervision based on random groups (SRG): This methodology allows us to isolate the performance benefits of selecting robots to form the target morphology based on their location in the environment. That is, we use the control strategy presented in LSM but disable the iterative growth process in control state STC. Instead, we repeat the iterative elimination process to select a group of randomly located foot-bots for self-assembly. The foot-bots do not have a priori knowledge about the task or the target morphology. The foot-bots initially move towards the light until the eye-bot starts transmitting messages. In case a gap wider than 5 cm is detected, the eye-bot establishes an STC link to a random foot-bot, i.e., without considering its location in the environment with respect to the gap. The eye-bot repeats this process until the number of foot-bots with established STC links matches the size of the target morphology. This group of randomly located foot-bots receive a SWARMORPH-script from the eye-bot containing the instructions necessary to form a target morphology that depends on the gap width. Once the morphology is formed, the foot-bots move towards the light to cross the gap.

7.3 Experiments and results

For each combination of gap width and control methodology, we ran 100 simulation runs (i.e., $4 \times 3 \times 100 = 1\,200$ runs in total). We first analyze the benefits of aerial supervision by comparing LSM with NCC. Then, we isolate the benefits of location-based group selection by comparing LSM with SRG. Videos of the experiments are available online [49].

7.3.1 NCC vs. LSM

We quantify the difference in task execution times between strategies NCC and LSM. The results are shown in Fig. 8a. We have only plotted the results of the narrowest gap of 5 cm for NCC, as the task completion times between the various gap widths did not prove to be significantly different. This is a direct consequence of the fact that the foot-bots executing this methodology formed chain morphologies of the same length regardless of the gap width they encountered.

In all the experiments, the foot-bots correctly performed the task. According to the results in Fig. 8a, the median task completion times of LSM are 51, 259 and 403 seconds for the widths 5 cm, 10 cm and 15 cm, respectively. Compared to the median task completion time of NCC (434 seconds), the mean completion times for LSM were respectively 88%, 40% and 7% lower in environments with gaps that can be crossed by an individual or chains composed of two or three foot-bots. This is due to the fact that in LSM, the length

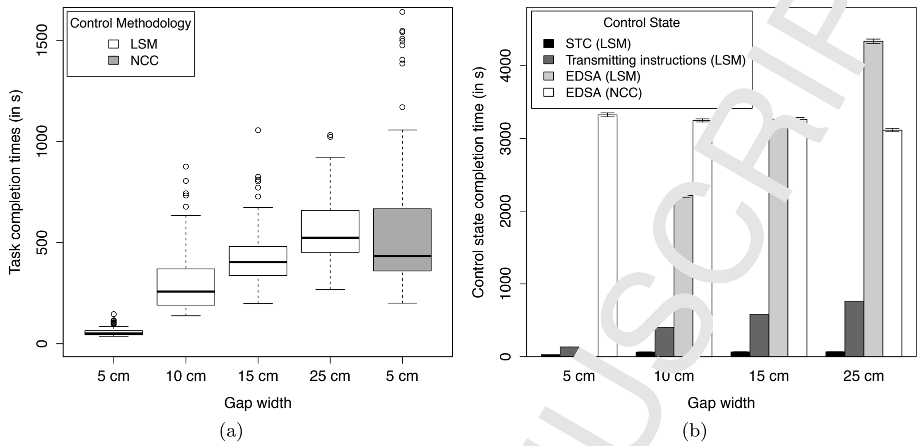


Figure 8: Results confirming how supervision from the eye-bot (i.e., LSM) lowers the task completion times of the overall system when compared to a control strategy (i.e., NCC) that does not rely on aerial supervision. (a) Box-and-whisker plot showing task completion times (in seconds) of LSM in four different environments. For the NCC methodology, results of only one environment is plotted as they do not differ significantly from each other. (b) Bar-plot showing a breakdown of the time spent by the foot-bots executing different control states. Bars are decorated with the standard deviation, except for the control state ”transmitting instructions” that is bound to constant time for each transmitted SWARMORPH-script (i.e., the gap width).

of the chain is chosen based on the gap width. The supervision provided by the eye-bot avoids the inclusion of excess foot-bots in the morphology requiring additional time for the formation of the target morphology. In the case of the widest gap (i.e., 25 cm) that can only be crossed by four or more physically connected foot-bots, NCC is, in general, faster than LSM. Intuitively, this could have been expected given that both control methodologies (i.e., LSM and NCC) form a chain of four foot-bots close to the gap, but in the case of LSM, self-assembly instructions need to be first transmitted from the eye-bot to the foot-bots before the self-assembly process can start. However, the NCC methodology has several outlier runs that take very long to complete. This is because in NCC, the foot-bots that become part of the target morphology are not pre-selected by the eye-bot. Hence, non-connected foot-bots can cause (sometimes severe) physical interference with ongoing self-assembly processes or with moving target morphologies. Both interferences delay task completion times.

In Fig. 8b we present a breakdown of how much time is spent by the foot-bots in each control state. As the results show, spatially targeted communication (STC, used in the LSM control methodology) is the control state that requires the least time, independent of gap size. More precisely, the completion times for STC were 2.6 s, 6.2 s, 6.5 s, and 6.6 s for selecting 1, 2, 3, and 4 foot-bots, respectively. In the case of selecting a single foot-bot (gap size 5 cm), the average value in simulation is higher than the average of 4.3 s from real robot experiments we presented separately in [19]. This is because, in simulation, robot control loops, and therefore vision updates, are available to aerial robots every 100 ms while the AR.Drone is able to retrieve and process images every 60 ms. We also observe similar results for the selection of 2 and 4 foot-bots with 5.1 s and 5.4 s, respectively. For more details on the performance of the STC control state in real robot experiments and a theoretical model that describes the scalability properties, we refer the reader to [19]. The results also show that the wider the gap, the more time is spent by the robots transmitting self-assembly instructions. This is due to the fact that the length of the SWARMORPH-script describing the target morphology grows linearly with the size of the target morphology. However, this communication overhead part of LSM would become negligible if a communication modality with higher bandwidth (such as wireless Ethernet) were used for communication. The results also show that when a target morphology composed of four foot-bots is formed, the self-assembly process in EDSA (LSM) requires on average 39% more time than that of EDSA (NCC). This can be explained by the fact that in NCC all foot-bots are available for forming a connection during the morphology growth process which increases the chances of a foot-bot being located close to where a connection is required causing connections to be formed faster. On the other hand, LSM selects neighboring foot-bots relative to the initially selected robot. However, LSM allocates precisely the number of resources required for self-assembly by selecting the required number of foot-bots needed for the target morphology and freeing up the rest of the team for other tasks. The decision involving this trade-off between faster target morphology formation times and more efficient resource allocation may depend on the task and mission priorities.

7.3.2 LSM vs. SRG

Here we isolate the performance gains that result immediately from the STC control state. For this purpose, we compare the results of the control methodologies LSM and SRG. As the selection of all foot-bots required for the target morphology uses different methods (LSM uses the iterative growth process to select a group of co-located foot-bots while SRG repeats the iterative elimination process to establish communication links to randomly located foot-bots), we only consider the time the eye-bot spent on selecting the foot-bot initiating the self-assembly process in control state STC.

As the results in Fig. 9 show, LSM was on average faster than SRG in all cases studied independent of the width of the gap. The explanation for these results is that a target morphology formed next to the gap by involving nearby foot-bots in most cases requires less time to finish the formation, and then reaches and crosses the gap faster than a morphology formed at a random place with peer foot-bots joining from random places in the environment. We expect this difference in task completion time to become even greater for larger start zones as the distances between randomly selected foot-bots and the gap would be larger.

8 Conclusions and future work

In this paper, we introduced supervised morphogenesis — a novel approach that enables aerial robots to provide assistance to ground-based self-assembling robots. We showed how this aerial assistance can help robots on the ground avoid costly self-assembly processes when they are not required. Furthermore, we showed that the presented control methodology can be used to enable the formation of appropriate morphologies without a priori knowledge of task and environment. A key feature of supervised morphogenesis is its high portability to other systems because it does not depend on proprietary hardware and can be implemented using standard cameras, LEDs, and wireless Ethernet-based communica-

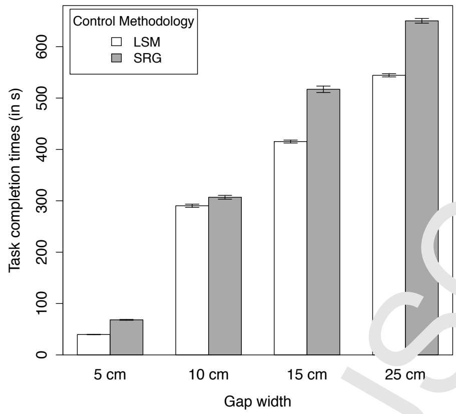


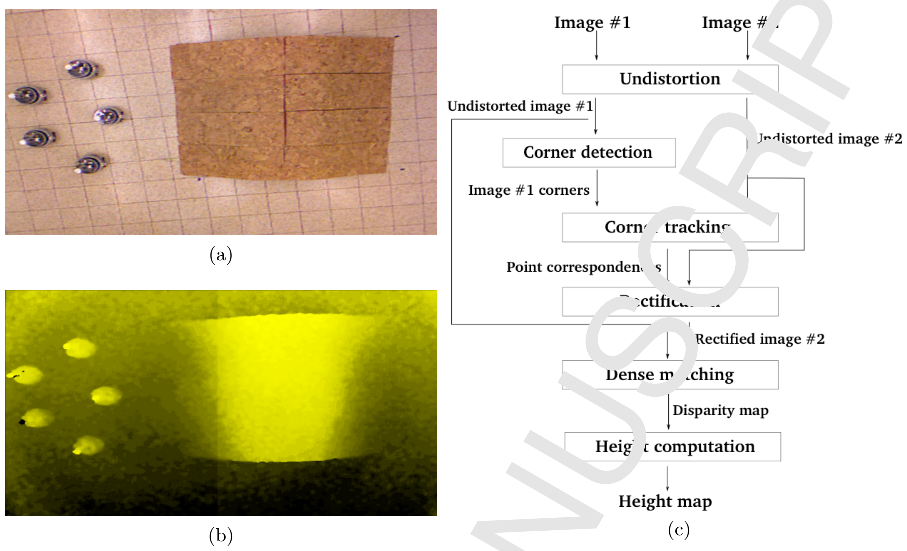
Figure 9: Results of the experiments carried out to isolate the benefits of group selection based on location (LSM) vs. random group selection (SRG). The results presented are task completion times of the two strategies for all the four environments considered. Note that the group selection and formation times have been omitted for LSM and SRG in order to facilitate a meaningful comparison. Standard deviations are added to the bars.

tion available to most robotic platforms. We showed how input from standard monocular cameras can be used to build two or three-dimensional models of the environment that allow aerial robots to perform on-board simulations. We reported on the results of two case studies we carried out in which supervised morphogenesis was demonstrated in two different heterogeneous teams with different sets of abilities. To the best of our knowledge, the work presented in this paper represents the first implementation of a robotic system that enables aerial robots to supervise self-assembly in ground-based robots. We showed that the presented control methodology for cooperation can provide performance benefits by enabling aerial robots to allocate the precise number of resources needed for a target morphology by recruiting robots based on their location on the ground and based on their mutual proximity.

One interesting direction for future work would be to study how aerial robots can provide supervision that enables the formation of different morphologies in parallel. Another interesting direction would be to study how target morphologies can be determined based on physics-based simulations for tasks that require solutions based on the physical characteristics of robots and objects in the environment or for tasks that have low levels of fault tolerance.

9 Acknowledgements

This work was supported by the European Research Council through the ERC Advanced Grant "E-SWARM: Engineering Swarm Intelligence Systems" (contract 246939) to Marco Dorigo. Nithin Mathews acknowledges support from Netcetera AG and from Wallonia-Brussels-International (WBI) through a Scholarship for Excellence grant. Anders Lyhne Christensen acknowledges support from Fundação para a Ciência e a Tecnologia (FCT) through grant UID/EEA/50008/2013. Marco Dorigo acknowledges support from the Funds for Scientific Research F.R.S.-FNRS of Belgium's French Community of which he is a research director.



(a) (b) (c)

Figure 10: Computing height maps using images retrieved from a monocular camera. (a) A 640x480 image acquired by the eye-bot. (b) A representation of the computed disparity map in which brighter pixels denote greater motion and lesser distance from the camera. (c) A flowchart showing the computation of height maps based on a pair of stereo images. The result is a two-dimensional matrix with the elevation in cm for each pixel.

Appendices

A Computing height maps

A.1 Method 1: monocular camera

We use two consecutively taken images returned by the eye-bot's downward pointing camera (see for an example Fig. 10a) to compute a height map of the surface and of the objects in the eye-bot's field of view. From two such images, each taken from different positions, the eye-bot first computes a *disparity map*, see Fig. 10b. For a pair of stereo images, a disparity map contains the distance (in pixel) by which each point in the first image has moved in the second image. For instance, the displacement of points closer to the camera is higher than that of the points further away from the camera. In a second step, the eye-bot calculates the height of each point in real-world distances based on the disparity of each point, the elevation (in cm) of the eye-bot, the displacement between the two images, and the properties of the camera. We summarize the individual steps required to compute a height map based on a pair of stereo image using the flowchart shown in Fig. 10c.

A.2 Method 2: the Kinect sensor

We obtain height maps of the eye-bot's field of view directly from a Microsoft Kinect sensor mounted on the eye-bot. All necessary computations are carried out by the sensor and height maps are available in almost real-time. The sensor does not require any prior

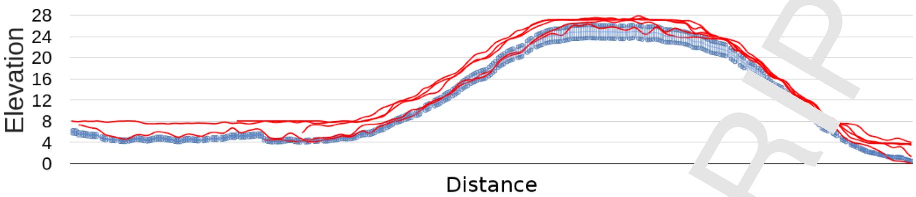


Figure 11: Mean elevation of height profiles acquired by the eye-bot from 10 different height maps. Graphs in red result from computation based on stereo images while data retrieved from the Kinect sensor is shown in blue.

knowledge of the environment and it can be operated under most light conditions. Despite the obvious advantages Kinect offers to increase the sensing capabilities of aerial robots, its weight (ca. 1.4 kg) can be seen as a disadvantage that reduces flight autonomy significantly for most application scenarios and existing aerial robots.

A.3 Quantitative comparison

In Fig. 11, we present a quantitative analysis of the data obtained with the two methods considered to acquire height maps. For this purpose, we compare the height profiles of all five foot-bot's estimated trajectory, i.e., a straight line connecting its position in the deployment area to the light source. The height profile is the mean elevation along an estimated trajectory and is computed from 10 different height maps computed on the basis of 10 sets of stereo images (plotted in red) or extracted from 10 different height maps returned by the Kinect (plotted in blue). In the latter case, we have only plotted the height profile of the longest foot-bot trajectory along with error intervals representing the standard deviation as the values of five profiles are too close to each other to be plotted in a clearly comprehensible manner. The standard deviation for the elevation computed using stereo images is 2.91 cm (not shown in the figure) as opposed to the 2.14 cm for the elevation retrieved from the Kinect indicating a slightly more reliable data source. However, as the figure shows, both methods deliver sufficiently precise estimates. While we have observed that in our setup the absolute values of the surface elevation computed using the stereo images constantly resulted in real-world values above those acquired from the Kinect and the ground truth, the relative differences between any two points is almost identical for the two methods. That is, we observed that the inclination computed between any two points in a height profile resulted in almost identical values independent of the underlying method. Note that the absolute values returned by neither method matches the ground truth (not shown) which remained between 3 cm and 4 cm under the values returned by the Kinect. This is clearly visible in Fig. 11 for the flat surface area to the left of the hill obstacle. One explanation may be the fact that the eye-bot was slightly tilted (and hence not parallel to the ground) when the data was collected.

References

[1] M. Dorigo, E. Tuci, V. Trianni, R. Groß, S. Nouyan, C. Ampatzis, T. H. Labella, R. O'Grady, M. Bonani, and F. Mondada. SWARM-BOT: Design and implementation of colonies of self-assembling robots. In *Computational Intelligence: Principles and Practice*, chapter 6, pages 103–135. IEEE Computational Intelligence Society, Piscataway, NJ, New York, 2006.

[2] H. Wei, Y. Chen, J. Tan, and T. Wang. Sambot: A self-assembly modular robot system. *IEEE/ASME Transactions on Mechatronics*, 16(4):745–757, 2011.

[3] L. Murray, J. Timmis, and A. Tyrrell. Self-reconfigurable modular e-pucks. In *Proceedings of the 8th International Conference on Swarm Intelligence*, volume 7461 of *LNCS*, pages 133–144. Springer, Berlin, Germany, 2012.

[4] M. Bonani, V. Longchamp, S. Magnenat, P. Rétornaz, D. Burnier, G. Roulet, F. Vaussard, H. Bleuler, and F. Mondada. The MarXbot, a miniature mobile robot opening new perspectives for the collective-robotic research. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4187–4193. IEEE Press, Piscataway, NJ, 2010.

[5] M. Rubenstein, A. Cornejo, and R. Nagpal. Programmable self-assembly in a thousand-robot swarm. *Science*, 345(6198):795–799, 2014.

[6] W. Liu and A. Winfield. Autonomous morphogenesis in self-assembling robots using IR-based sensing and local communications. In *Proceedings of the 7th International Conference on Swarm Intelligence*, volume 6234 of *LNCS*, pages 107–118. Springer, Berlin, Germany, 2010.

[7] A. L. Christensen, R. O'Grady, and M. Dorigo. SWARMORPH-script: a language for arbitrary morphology generation in self-assembling robots. *Swarm Intelligence*, 2(2–4):143–165, 2008.

[8] E. Klavins, R. Ghrist, and D. Lipsky. A grammatical approach to self-organizing robotic systems. *IEEE Transactions on Automatic Control*, 51(6):949–962, 2006.

[9] H. Wei, Y. Chen, M. Liu, Y. Cai, and T. Wang. Swarm robots: From self-assembly to locomotion. *The Computer Journal*, 54(9):1465–1474, 2011.

[10] R. O'Grady, A. L. Christensen, C. Pinciroli, and M. Dorigo. Robots autonomously self-assemble into dedicated morphologies to solve different tasks. In *Proceedings of 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1517–1518. IFAAMAS, Richland, SC, 2010.

[11] S. L. Waslander. Unmanned aerial and ground vehicle teams: recent work and open problems. In *Autonomous Control Systems and Vehicles: Intelligent Unmanned Systems*, pages 21–36. Springer, Tokyo, Japan, 2013.

[12] S. Lacroix and G. Besnerais. Issues in cooperative air/ground robotic systems. In *Robotics Research*, volume 66 of *Springer Tracts in Advanced Robotics*, pages 421–432. Springer, Berlin, Germany, 2011.

[13] V. Kumar and N. Michael. Opportunities and challenges with autonomous micro aerial vehicles. *International Journal of Robotics Research*, 31(11):1279–1291, 2012.

[14] C. Bills, J. Chen, and A. Saxena. Autonomous MAV flight in indoor environments using single image perspective cues. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5776–5783. IEEE Computer Society Press, Los Alamitos, CA, 2011.

[15] S. Zingg, D. Scaramuzza, S. Weiss, and R. Siegwart. MAV navigation through indoor corridors using optical flow. In *IEEE International Conference on Robotics and*

Automation (ICRA), pages 3361–3368. IEEE Computer Society Press, Los Alamitos, CA, 2010.

[16] S. Shen, N. Michael, and V. Kumar. Autonomous multi-floor indoor navigation with a computationally constrained MAV. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 20–25. IEEE Computer Society Press, Los Alamitos, CA, 2011.

[17] A. Briod, P. Kornatowski, J.-C. Zufferey, and D. Floreano. A collision-resilient flying robot. *Journal of Field Robotics*, 31(4):496–509, 2014.

[18] A. Briod, A. Klaptocz, J. C. Zufferey, and D. Floreano. The Airburr: A flying robot that can exploit collisions. In *ICME International Conference on Complex Medical Engineering (CME)*, pages 569–574. IEEE Press, Piscataway, NJ, 2012.

[19] N. Mathews, G. Valentini, A. L. Christensen, R. O'Grady, A. Brutschy, and M. Dorigo. Spatially targeted communication in decentralized multirobot systems. *Autonomous Robots*, 38(4):439–457, 2015.

[20] N. Mathews, A. L. Christensen, R. O'Grady, P. Rétornaz, M. Bonani, F. Mondada, and M. Dorigo. Enhanced directional self-assembly based on active recruitment and guidance. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4762–4769. IEEE Computer Society Press, Los Alamitos, CA, 2011.

[21] N. Mathews, A. L. Christensen, R. O'Grady, and M. Dorigo. Cooperation in a heterogeneous robot swarm through spatially targeted communication. In *Proceedings of the 7th International Conference on Swarm Intelligence (ANTS)*, volume 6234 of *LNCS*, pages 400–407. Springer, Berlin, Germany, 2010.

[22] R. O'Grady, A. L. Christensen, C. Pinciroli, and M. Dorigo. Robots autonomously self-assemble into dedicated morphologies to solve different tasks (extended abstract). In *9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1517–1518. IFAAMAS, Richland, SC, 2010.

[23] N. Mathews, A. L. Christensen, R. O'Grady, F. Mondada, and M.Dorigo. Mergeable nervous systems for robots. *Nature Communications*, 8(439), 2017.

[24] S. Lacroix, I.-K. Jung, and A. Mallet. Digital elevation map building from low altitude stereo imagery. *Robotics and Autonomous Systems*, 41(2–3):119–127, 2002.

[25] C. Forster, M. Faessler, F. Fontana, M. Werlberger, and D. Scaramuzza. Continuous on-board monocular-vision-based elevation mapping applied to autonomous landing of micro aerial vehicles. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 111–118. IEEE Computer Society Press, Los Alamitos, CA, 2015.

[26] N. Michael, S. Shen, K. Mohta, Y. Mulgaonkar, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake, K. Yoshida, K. Ohno, E. Takeuchi, and S. Tadokoro. Collaborative mapping of an earthquake-damaged building via ground and aerial robots. *Journal of Field Robotics*, 29(5):832–841, 2012.

[27] J. H. Kim, J. W. Kwon, and J. Seo. Multi-UAV-based stereo vision system without GPS for ground obstacle mapping to assist path planning of UGV. *Electronics Letters*, 50(20):1431–1432, 2014.

[28] M. A. Montes de Oca, E. Ferrante, N. Mathews, M. Birattari, and M. Dorigo. Opinion dynamics for decentralized decision-making in a robot swarm. In *Proceedings of the Seventh International Conference on Swarm Intelligence (ANTS 2010)*, pages 252–263. Springer-Verlag, Berlin, Germany, 2010.

[29] C. Luo, A. P. Espinosa, D. Pranantha, and A. De Gloria. Multi-robot search and rescue team. In *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 296–301. IEEE Press, Piscataway, NJ, 2011.

[30] M. Dorigo, D. Floreano, L. M. Gambardella, F. Mondada, S. Nolfi, T. Baaboura, M. Birattari, M. Bonani, M. Brambilla, A. Brutschy, D. Burnier, A. Campo, A. L. Christensen, A. Decugnière, G. Di Caro, F. Ducatelle, E. Ferrante, A. Förster, J. Guzzi, V. Longchamp, S. Magnenat, J. Martinez Gonzales, N. Mathews, M. Montes de Oca, R. O'Grady, C. Pinciroli, G. Pini, P. Rétornaz, J. Roberts, V. Sperati, T. Stirling, A. Stranieri, T. Stützle, V. Trianni, E. Tuci, A. E. Turgut, and F. Vaussard. Swarmanoid: A novel concept for the study of heterogeneous robotic swarms. *IEEE Robotics & Automation Magazine*, 20(4):60–71, 2013.

[31] M. Langerwisch, M. Ax, S. Thamke, T. Remmersmann, A. Tiderko, K.-D. Kuhnert, and B. Wagner. Realization of an autonomous team of unmanned ground and aerial vehicles. In *5th International Conference on Intelligent Robotics and Applications (ICIRA)*, pages 302–312. Springer, Berlin, Germany, 2012.

[32] M. Garzón, J. Valente, D. Zapata, and A. Barrientos. An aerial-ground robotic system for navigation and obstacle mapping in large outdoor areas. *Sensors*, 13(1):1247–1267, 2013.

[33] M. Persson, T. Duckett, and A.J. Lilienthal. Fusion of aerial images and sensor data from a ground vehicle for improved semantic mapping. *Robotics and Autonomous Systems*, 56(6):483–492, 2008.

[34] E. Mueggler, M. Faessler, F. Fontana, and D. Scaramuzza. Aerial-guided navigation of a ground robot among movable obstacles. In *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 1–8. IEEE Press, Piscataway, NJ, 2014.

[35] M.B. Mosely, B.P. Grocholsky, C. Cheung, and S. Singh. Integrated long-range UAV/UGV collaborative target tracking. In *Proceedings of SPIE, Unmanned Systems Technology XI Conference*, volume 7332. SPIE Press, Bellingham, WA, 2009.

[36] R. Käslin, P. Fankhauser, Z. Stumm, E.and Taylor, E. Mueggler, J. Delmerico, D. Scaramuzza, R. Siegwart, and M. Hutter. Collaborative localization of aerial and ground robots through elevation maps. ETH Zurich, Zurich, Switzerland, 2016.

[37] P. Pace, G. Aloi, G. Caliciuri, and G. Fortino. A mission-oriented coordination framework for teams of mobile aerial and terrestrial smart objects. *Mobile Networks and Applications*, 21(4):708–725, 2016.

[38] M. Saska, V. Vonásek, T. Krajník, and L. Přeučil. Coordination and navigation of heterogeneous mav-ugv formations localized by a "hawk-eye"-like approach under a model predictive control scheme. *International Journal of Robotics Research*, 33(10):1393–1412, 2014.

[39] E. H. C. Harik, F. Guinand, H. Pelvillain, F. Guérin, and J. F. Brethé. A decentralized interactive architecture for aerial and ground mobile robots cooperation. In *International Conference on Control, Automation and Robotics (ICCAR)*, pages 37–43. IEEE Press, Piscataway, NJ, 2015.

[40] P.-J. Bristeau, F. Callou, D. Vissière, and N. Petit. The navigation and control technology inside the AR.Drone micro UAV. In *Proceedings of the 18th IFAC World Congress*, pages 1477–1484. IFAC-PapersOnLine, Centerville, OH, 2011.

[41] T. Krajník, V. Vonásek, D. Fišer, and J. Faigl. AR-Drone as a platform for robotic research and education. In *Research and Education in Robotics (EUROBOT)*, volume 161, pages 172–186. Springer, Berlin, Germany, 2011.

[42] J.F. Roberts. *Enabling collective operation of indoor flying robots*. PhD thesis, EPFL, École polytechnique fédérale de Lausanne, Lausanne, Switzerland, 2011.

[43] J.F. Roberts, T. Stirling, J.-C. Zufferey, and D. Floreano. 3-D relative positioning sensor for indoor flying robots. *Autonomous Robots*, 33(1-2):5–20, 2012.

[44] J.F. Roberts, T.S. Stirling, J-C. Zufferey, and D. Floreano. 2.5D infrared range and bearing system for collective robotics. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3659–3664. IEEE Press, Piscataway, NJ, 2009.

[45] N. Mathews, A. L. Christensen, E. Ferrante, R. O'Grady, and M. Dorigo. Establishing spatially targeted communication in a heterogeneous robot swarm. In *Proceedings of 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 939–946. IFAAMAS, Richland, SC, 2010.

[46] F. Mondada, L. M. Gambardella, D. Floreano, S. Nolfi, J.-L. Deneubourg, and M. Dorigo. The cooperation of swarm-bots: Physical interactions in collective robotics. *IEEE Robotics & Automation Magazine*, 12(2):21–28, 2005.

[47] N. Mathews, A. Stranieri, A. Scheidler, and M. Dorigo. Supervised morphogenesis – morphology control of ground-based self-assembling robots by aerial robots. In *Proceedings of 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 97–104. IFAAMAS, Richland, SC, 2012.

[48] N.Mathews, A. L. Christensen, R. O'Grady, and M. Dorigo. Spatially targeted communication and self-assembly. In *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2678–2679. IEEE Computer Society Press, Los Alamitos, CA, 2012.

[49] Online supplementary material: Supervised morphogenesis: towards reaching the potential of self-assembling robots through cooperation with aerial robots. http://iridia.ulb.ac.be/supp/IridiaSupp2017-007.

[50] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in computational stereo. *Transactions on Pattern Analysis and Machine Intelligence*, 25:993–1008, 2003.

[51] 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*, 6(4):271–295, 2012.

Alessandro Stranieri received the B.Eng. and the M.Eng. in Computing Systems Engineering from Politecnico di Milano. Between 2010 and 2012 he collaborated on several robotics projects at the IRIDIA laboratory, Department of Applied Sciences at the Free University of Brussels, Belgium. Since 2013 he has pursued a carreer in Software Development and he currently works as a Software Engineer in Brussels, Belgium.

Alexander Scheidler received the Diploma and Ph.D. degrees in computer science from the University of Leipzig, Leipzig, Germany, in 2005 and 2010, respectively. He was a Post-Doctoral Fellow with IRIDIA, UniversitÃ© Libre de Bruxelles, Brussels, Belgium, from 2010 to 2012 and has been with the Fraunhofer Institute for Wind Energy and Energy System Technology, Kassel, Germany, since 2012. His current research interests include swarm intelligence, optimization methods for distribution grid planning, and HPC in power system research.


ACCEPTED MANUSCRIPT

Anders Lyhne Christensen received a Ph.D. degree from IRIDIA-CoDE, UniversitÃ© Libre de Bruxelles, Belgium in 2008, and he is currently an assistant professor at University Institute of Lisbon (ISCTE-IUL), Portugal. Dr. Christensen is the founder and head of the Bio-inpsired Computation and Intelligent Machines Lab at ISCTE-IUL. His research interests include bio-inspired machine intelligence, large-scale multirobot systems, evolutionary computation, and fault tolerance.

ACCEPTED MANUSCRIPT

Marco Dorigo received his PhD degree in electronic engineering in 1992 from the Politecnico di Milano, Milan, Italy, and the title of Agr{\'e}g{\'e} de l'Enseignement Sup{\'e}rieur, from ULB, in 1995.

From 1992 to 1993, he was a Research Fellow at the International Computer Science Institute, Berkeley, CA. In 1993, he was a NATO-CNR Fellow, and from 1994 to 1996, a Marie Curie Fellow. Since 1996, he has been a tenured Researcher of the FNRS, the Belgian National Funds for Scientific Research, and co-director of IRIDIA, the artificial intelligence laboratory of the ULB. He is the inventor of the ant colony optimization metaheuristic. His current research interests include swarm intelligence, swarm robotics, and metaheuristics for discrete optimization. He is the Editor-in-Chief of Swarm Intelligence, and an Associate Editor or member of the Editorial Board of many journals on computational intelligence and adaptive systems.

Dr. Dorigo is a Fellow of AAAI, ECCAI and IEEE. He was awarded the Italian Prize for Artificial Intelligence in 1996, the Marie Curie Excellence Award in 2003, the Dr. A. De Leeuw-Damry-Bourlart award in applied sciences in 2005, the Cajastur International Prize for Soft Computing in 2007, an ERC Advanced Grant in 2010, the IEEE Frank Rosenblatt Award in 2015, and the IEEE Evolutionary Computation Pioneer award in 2016.

Nithin Mathews holds a Ph.D. degree from the UniversitÃ© Libre de Bruxelles (Belgium), an M.Sc. degree from the University of Freiburg (Germany), and a Dipl.-Ing degree from the University of Applied Sciences and Arts Northwestern Switzerland (Switzerland). Both his Ph.D. and M.Sc. studies were focused on the subject of artifical intelligence and robotics. His research emcompasses the fields of swarm robotics, self-assembling robots, and air-ground robot teams. Since 2016, he is also an employee of Netcetera AG, Switzerland.

ACCEPTED MANUSCRIPT