Insect-Inspired Robot Coordination: Foraging and Coverage

Sjriek Alers[1], Karl Tuyls[2], Bijan Ranjbar-Sahraei[1], Daniel Claes[2] and Gerhard Weiss[1]

[1]Maastricht University, P.O. Box 616, 6200 MD, Maastricht, The Netherlands
[2]University of Liverpool, Ashton Building, Liverpool L69 3BX, United Kingdom

Abstract

In this paper we investigate coordination principles inspired by the behaviour of honeybees and ants for coordination purposes in multi-robot systems. Specifically, we study the problem instances of bee-inspired robot Foraging and ant-inspired robot Coverage, where Foraging is the problem of exploring the environment in search of food or provisions and Coverage is the problem of deploying a robotic swarm in the environment with the task of maximising the sensor coverage of the environment. To effectively and efficiently solve both problems, distributed multi-robot coordination is required. For the first problem we investigate a bee-inspired solution method. The second problem is studied using a stigmergic approach. In an extensive set of experiments we first study the feasibility of the proposed multi-robot coordination for robotic swarms with extended resources and discuss the benefits and limitations of using these swarms. Furthermore, as the downsizing in swarm robotics becomes increasingly important with ongoing miniaturization in various applications, the feasibility of the proposed coordination techniques for robotic swarms with limited resources is studied in detail; the practical requirements for overcoming the limitations of these swarms are introduced and the main need to incorporate these robots in real world experiments is discussed.

Introduction

Swarm robotic systems are motivated by a wide range of application areas, such as for instance surveillance and patrolling, where mobile guarding robots are considered as an alternative and improvement over fixed security cameras and even humans. Other application areas include exploration and identification of hazardous environments (e.g., nuclear plants and fire detection), mobile sensor networks, space exploration, etc.

Recent years have seen an increasing interest in taking inspirations from natural phenomena for solving computational problems in disciplines at the intersection of computer science, robotics and economics. An interesting natural phenomenon is the behaviour that can be observed in colonies of social insects such as ants and bees. For instance, recent work shows a strong potential in creating artificial systems that mimic insect behaviour that can solve complex coordination tasks such as e.g., routing on the internet, mobile ad hoc network routing, robotic tasks, etc. (Lemmens and Tuyls, 2012; Dressler and Akan, 2010; Floreano and Mattiussi, 2008). These insects have evolved over a long period of time and display remarkable behaviours that are highly suitable for addressing the complex tasks that they are facing. Swarm optimisation algorithms, like ant colony optimisation (Dorigo et al., 2006), rely on pheromone trails to mediate (indirect) communication between agents.

These pheromones need to be deposited and sensed by agents while they decay over time. Though easy to simulate, artificial pheromones are hard to bring into real-life robotic applications. However, recently non-pheromone-based algorithms where developed as well (Lemmens, 2011). Such algorithms are inspired by the foraging and nest-site selection behaviour of (mainly) bees. In general, bees explore the environment in search for high quality food sources and once returned to the hive they start to dance in order to communicate the location of the source. Using this dance, bees recruit other colony members for a specific food source.

We draw inspiration from these insect behaviours with the goal to create emergent intelligent systems for distributed coordination that can be deployed in real world settings. However, doing the experiments with real robotic swarms is very challenging. On one side robotic swarms with limited resources such as e-pucks (Mondada et al., 2009) are robust, compact, easy to setup and relatively cheap in terms of price but have limited sensing, computation and actuation capabilities. On the other hand robotic swarms with extended resources, which contain advanced cameras and general purpose computers, have high computational power, good movement capabilities and are modular. In this paper, we provide an extensive study of the pros and cons of each of these swarm types.

The remainder of the paper is organised as follows. First we introduce the biological background of the foraging and coverage approaches. Then we continue with the swarm robotics approach with extended resources and it is explained how multi-robot coordination can be implemented in these swarms in a set of experiments, highlighting the vision and communication capabilities of such advanced robots.

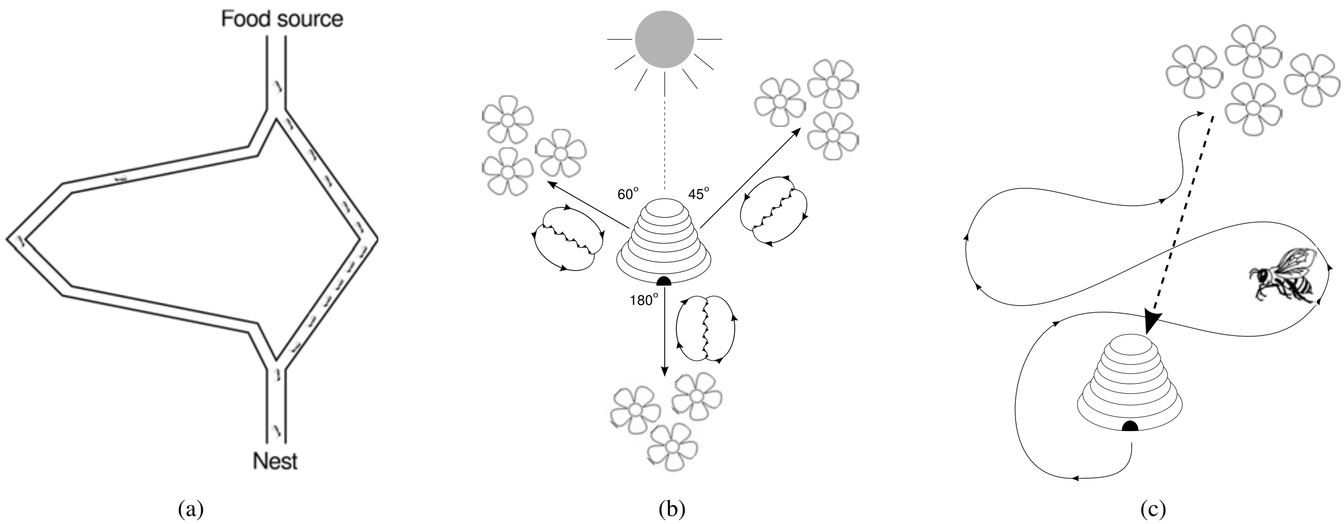


Figure 1: Biological inspiration (a) ants exploring two paths; the shortest path prevails. (b) honeybee waggle dance communicating a PI vector. (c) Lévy flight and path integration.

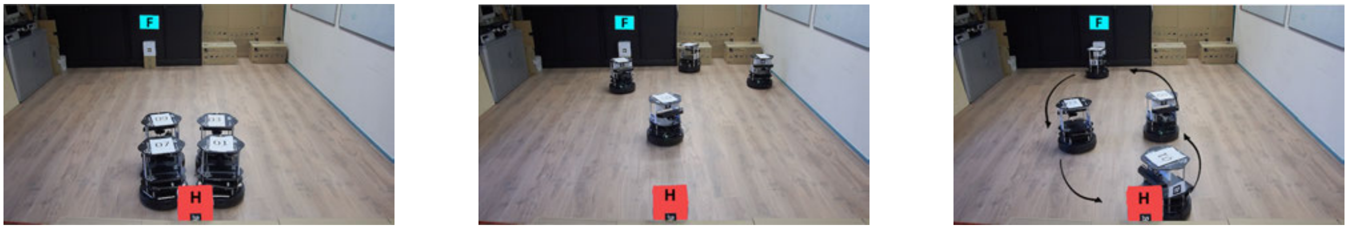
After that we continue by elaborating the coordination approaches for swarm robotics with limited resources: vision and communication techniques are explained; implementation of both multi-robot coverage and foraging are presented in a set of experiments in the context of limited available resources, highlighting the novel techniques in computation and communication that are used to make such implementations possible. Finally we conclude.

Biological Inspiration

Most of the research in swarm intelligence revolves around the behaviour of ants (Dorigo and Stützle, 2004; Dorigo and Blumb, 2005; Dorigo et al., 2006). The principle is simple yet elegant: ants deposit a pheromone trail on the path they take during travel. Using this trail, they are able to navigate toward their nest or food and communicate with their peers. More specifically, ants employ an indirect recruitment strategy by accumulating pheromone trails. When a trail gets strong enough, other ants are attracted to it and will follow this trail toward a food destination. The more ants follow a trail, the more pheromone is accumulated and in turn the trail becomes more attractive for being followed. This is known as the autocatalytic process. Since long paths take more time to traverse, it will require more ants to sustain a long path. As a consequence, short paths will eventually prevail, see Figure 1(a). Pheromone-based algorithms are already used to address various problems successfully, such as (amongst others) the Traveling Salesman Problem (Dorigo and Stützle, 2004), Routing Problem (Di Caro et al., 2005), Group Shop Scheduling (Blum and Sampels, 2004), and area coverage with robots (Wagner et al., 1999; Ranjbar-Sahraei et al., 2012b).

Foraging honeybees display two types of behaviour, i.e., recruitment and navigation. In order to recruit other colony members for food sources, honeybees inform their nest mates of the distance and direction of these food sources by means of a waggling dance performed on the vertical combs in the hive. This dance (i.e., the bee language) consists of a series of alternating left-hand and right-hand loops, interspersed by a segment in which the bee waggles her abdomen from side to side. The duration of the waggle phase is a measure of the distance to the food. The angle between the sun and the axis of a bees waggle segment on the vertical comb, represents the azimuthal angle between the sun and a target location, i.e., the direction in which a recruit should fly (see Figure 1(b) and 1(c)). Other members of the colony can adopt the "advertisement" for a food source. The decision mechanism for adopting an "advertised" food-source location by a potential recruit is not completely understood. It is considered that the recruitment amongst bees is a function of the quality of the food source.

Different species of social insects, such as honeybees and desert ants, make use of non-pheromone-based navigation. Non-pheromone-based navigation mainly consists of Path Integration (PI), which is the continuous update of a vector by integrating all angles steered, and all distances covered (Collett et al., 1998). A PI vector represents the insects knowledge of direction and distance towards its destination. To construct a PI vector, the insect does not use a mathematical vector summation, but employs a computationally simple approximation (Collett et al., 1998). Using this approximation, the insect is able to return to its destination directly. More precisely, when the path is unobstructed, the insect solves the problem optimally. However, when the path is obstructed, the insect has to fall back on other strategies such as exploration or landmark navigation (Cheng et al., 1987;



(a) (b) (c)

Figure 2: Multi-Robot foraging using swarm robots with extended resources. (a) All robots start at the hive (H) location. (b) Robots are exploring the unknown environment randomly. The left two robots have found the food (F) location and are foraging between the hive and the food location. (c) All robots have converged to foraging behavior.

Collett et al., 2002) to solve the problem. Obviously, bees are able to fly, i.e., when they encounter an obstacle, they can mostly choose to fly over it. However, even if the path is unobstructed, bees tend to navigate over the entire path using landmarks. The landmarks divide the entire path into segments and each landmark has a PI vector associated with it. This behaviour decreases navigation errors and ensures robustness. We refer to a home-pointing PI vector as a Home Vector (HV). PI is used in both exploration and exploitation. During exploration insects constantly update their HV. It is however, not used as an exploration strategy. During exploitation, the insects update both their HV and the PI vector indicating the food source, and use these vectors as guidance to a destination.

Swarm Robotics with Extended Resources

In this section we introduce the swarm robotics with extended resources. These swarms use general purpose computers, high quality and advanced video cameras, 3D sensors for mapping (e.g., laser range finders), accurate wheel encoders that makes enhanced odometry possible, fused data of accelerometers, and a gyroscope.

The Turtlebot[1] platform is a robot with extended resources. This robot is equipped with a laptop with core-i3 CPU for computation that is running the Robot Operating System[2] framework.

As a main sensing unit the Turtlebot is equipped with a Kinect sensor. The full RGBD information is used to detect and locate AR markers. For static obstacle detection, we only use the depth information of the sensor together with three bumpers that are located in the front half of the robot. Furthermore, the robot estimates its position by integrating the wheel odometry and gyro information. Hence, no map of the environment is built and the only known reference point is the target location marker. This can lead to the problem that if the odometry is faulty, the robot does not always find the target location back. As a solution the robots fall back into a search mode, if this is the case. Another solution could be to implement a Northstar like navigation system, by providing a fixed frame of reference which is almost always visible from any location.

Robot Vision and Communication

To enable visual robot-robot detection we equipped every Turtlebot with six unique markers, which are oriented in a way that at least one marker is visible from any angle. To track and decode these markers we make use of a toolkit called ALVAR, more specifically we use the ROS wrapper[3] of this library. We use a customised bundle detection method to determine the center of the detected robot dependent on the decoded markers. Kalman filtering is applied to get better and more stable readings and consequently a more accurate estimate of the detected robots position, heading and speed. These parameters are used again for collision avoidance.

Communication is realised over Wi-Fi with a UDP connection to each Turtlebot using the LCM library[4]. Even though global communication would be possible, we limit the communication, such that every robot listens only to its own channel. To simulate local communication, the robots can only communicate with another robot when it is in view and in close proximity, i.e., less than one meter away.

Experiments

In this subsection we briefly describe the practical algorithms needed for implementation of bee foraging on swarms with extended resources. This clearly highlights the main benefit of using these type of robots which is the possibility of implementation of very advanced algorithms in a very convenient way.

Collision Avoidance In order to avoid robot to robot collisions, we rely on the marker detection to predict positions and speeds of the other robots. This information can be used to efficiently compute a non-colliding speed vector as we

[1]http://www.Turtlebot.com/
[2]http://www.ros.org/
[3]http://wiki.ros.org/ar_track_alvar
[4]https://code.google.com/p/lcm/

have developed previously in (Claes et al., 2012). In contrast to this previous approach, in which the robot-robot detection was avoided by using a global reference frame and broadcasting the positions to all robots via Wi-Fi, solely the marker detection and the predictions using a Kalman filter are used. This means that a few collisions still might occur due to failure to detect the markers of the other robots and additionally, there are certain configurations in which the robots cannot see each other due to the field of view of the Kinect sensor, e.g., when two robots drive in a V-shape towards each other, the field of view of the Kinect is too narrow to detect the other robot.

As shown in the previous work of authors in (Alers et al., 2014), multiple Turtlebots perform a foraging task, i.e., starting at the Hive (H) location and randomly exploring the unknown environment for a specific Food (F) location. This is shown in Figure 2. Another way of locating a food location is by asking bypassing robots for a known food location, which is done by simulating local communication over Wi-Fi. When the source is found the robot starts to exploit this source, i.e. driving from the food to the hive location until the food is depleted or a better source is found. A video showing this demonstration can be found in the online material[5].

Discussion

In summary, the robotic swarms with extended resources can accomplish many tasks successfully. These swarms use more sophisticated sensors like RGBD and VGA camera's to detect environmental features and can communicate with each other using Wi-Fi, while they use vision processing to simulate local communication.

Robots with extended resources have their up and downside. The downside of these robots are that they are bigger than the robots with limited resources; they are more expensive, as the sensors and computational units are more costly. These type of robots run on an operating system that has a steep learning curve. On the other hand, the upside of such robots is that they are more versatile. It is much easier to extend the platform with new sensors by for example plugging in an additional camera, or a dedicated control unit into the usb port. The computational limitations are not restrictive, e.g., image processing can be easily done without exhausting any other resource. Last but not least, the software modules can be easily reused or shared with the robotic community, as all the modules are developed in a standardized way.

Next we study the swarms of robots with limited resources which are simple, compact, small and relatively cheap. These robots are very robust by using lots of proven technologies (e.g., microcontrollers, basic sensors and actuators)

[5] `http://swarmlab.unimaas.nl/papers/aamas-2014-foraging`

Table 1: E-puck technical specification

Element	Technical information
Processor	dsPIC30F6014A @ 60 MHz (15 MIPS), 16-bit microcontroller with DSP core
Memory	RAM: 8KB Flash: 144 KB
Motors	2 stepper motors with a 50:1 reduction gear
Camera	VGA color camera with resolution of 640×480 pixels
LEDs	8 red LEDs on the ring, green LEDs on the body, 1 high intensity red LED in the front
Wireless Communication	Bluetooth for robot-computer and robot-robot communications, Infrared for robot-robot communication

Swarm Robotics with Limited Resources

In this section we introduce the swarm robotics with limited resources approach, which refers to the swarms of relatively small and cheap robots that have limited computation power (i.e., embedded micro-controllers), limited memory and very simple sensors and basic communications. Such swarms are in contrast to the swarms with extended resources that take advantage of powerful computers, advanced cameras and Wi-Fi communication capabilities.

The e-puck platform is an example of a robot with limited resources. e-puck is a small robot for educational and research purposes, developed by the EPFL University (Mondada et al., 2009). This robot is efficiently used in numerous projects in the domain of swarm robotics and swarm intelligence (e.g., works by Alers et al. (2011, 2013a,b); Lemmens et al. (2011); Breitenmoser et al. (2010); Mondada et al. (2009); Ranjbar-Sahraei et al. (2013b)). The main features of the e-puck robot include, but are not limited to; a robust design, flexibility for a large spectrum of educational activities, compact size, and rich on-board accessibilities (e.g., microphones, accelerometer and camera).

The e-puck hardware consists of different sensor types for detecting visible or Infra Red (IR) light, sound, acceleration, etc. The motors are the only actuators which are available in e-puck. A microprocessor of PIC family with 8 KB RAM memory assists the robot to get data from its sensors, analyse it, and perform actions. The main hardware elements of the e-puck robot are listed in Table 1.

As listed in the table, the on-board camera of the e-puck has a resolution of 640×480 pixels, although due to the limited resources, the robot is only capable of storing and processing an image of 40×40 pixels.. It is placed at the front of the e-puck, 2.7 cm above the floor. With this camera, objects that are placed on the floor can be detected at a minimum distance of 7.4 cm. The camera angle is approximately $40°$, and at this minimum distance, objects of 5.1 cm

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

Figure 3: Detectable features presented in (Alers et al., 2013b) (a) Landmarks with barcode. (b) QR-code level 3. (c) Robot LEDs. (d) Robot orientation pattern.

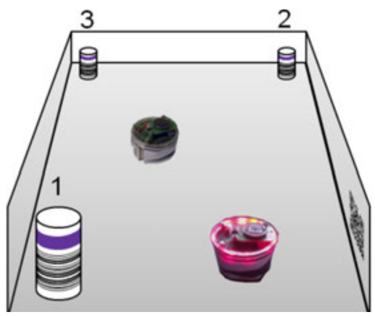


Figure 4: Scenario for validation of robotic vision.

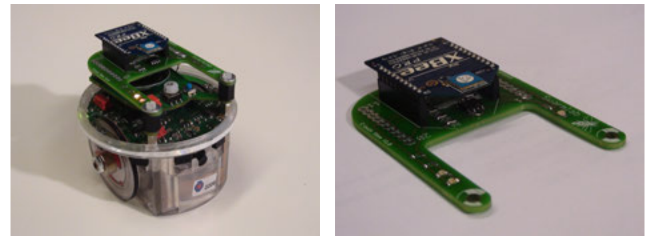
width can be fully monitored.

Robot Vision

Alers et al. (2013b) explored several visual features that can be used for acquiring information from the environment by a robot with limited computational abilities. In this work, for detecting key locations in the environment, such as corners in a maze, the usage of specific landmarks for these locations is investigated. Each landmark consists of an upper ring with a solid color, so that it can be detected from a distance, and on the lower part a unique barcode for keeping track of the landmark numbers, as can be seen in Figure 3(a).

Furthermore, the possibility to detect markers with an even higher data density, QR-codes as in Figure 3(b), are explored. The challenge in the detection of these two-dimensional codes, lies in analysing and processing the camera data with the limited processing and memory resources that are available in the robotic platform.

Finally, the most common feature already available in every swarm robotic setting is explored: the presence of an other robot. It's always favourable to detect the relative distance and orientation to other robots in respect of one's own position. Therefore, the available LEDs on the robot provide a very good feature for robot detection from a distance, see Figure 3(c). Moreover, a specific gradient pattern for nearby robot detection, as shown in Figure 3(d), is designed. This pattern results in a very accurate orientation and distance detection.



(a) (b)

Figure 5: e-puck robot with improved local communication (a) An e-puck robot as used in the experiments. (b) The XBee communication extension board.

A video of this performed experiment on a validation scenario as shown in Figure 4 can be found online[6], including the intermediate image data from the robot.

Communication

Direct Communication The e-puck robots contain 4 types of sensors with which they have to make sense of the world around them. Namely, 8 IR sensors, 1 camera, 3 microphones, and an accelerometer. Of these sensors, we only use the 8 IR sensors for proximity detection. Additionally, they have two main communication possibilities, namely, IR communication and (limited) Bluetooth communication. The former is prone to interference, its operation is CPU intensive, and as such is only viable for local short-message communication. The latter is limited to 3 simultaneous connections and setting up a single connection can take as long as 10 s. Moreover, 3 simultaneous connections can only be set up in the form of 1 master and 3 slaves. This severely limits communication possibilities.

In order to overcome the shortcomings of the current sensors, we were inspired by the communication module on the AdMoVeo robot designed by Alers and Hu (2009) and have designed an XBee extension board for the e-puck to improve local communication. Its design features are robustness, speed, low power usage, and ease of use. Figures 5(a) and 5(b) shows the e-puck equipped with the extension board and the extension board alone, respectively. XBee ensures reliable, fast, local, peer to peer communication. Moreover, it also provides the possibilities for creating a mesh network between multiple XBee chips. This opens up research possibilities in fields such as Mobile Adhoc Routing.

Stigmergic Communication Inspired by the stigmergic type of communication in ant colonies, robots can get benefits of stigmergy in communication-limited environments. However, despite of a few reports of using chemicals or radio frequency identification tags in robotic experiments by

[6] `http://swarmlab.unimaas.nl/papers/adaptive-2013-demo`

Figure 6: Darkroom with glow-in-the-dark floor, where the e-puck robots circle around and emit UV light onto the floor.

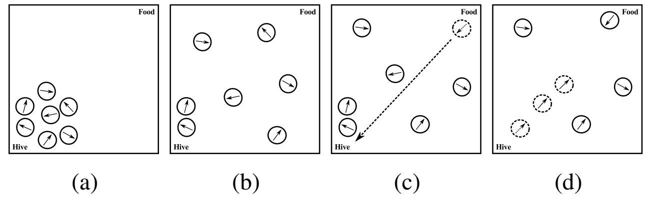


Figure 7: The four stages of Biomimicry Foraging (a) All robots start at the nest location. (b) The robots randomly disperse trough the environment looking for a food location. (c) A robot that has found food returns to the nest location by the shortest possible path. (d) The food location is communicated to other robots and they start to exploit this food-source.

Fujisawa et al. (2008); Herianto et al. (2007); Johansson and Saffiotti (2009), due to difficulties in implementation and limited extendibility, this approach did not provide sufficient applicability in swarm scenarios. Therefore, motivated by the technique proposed by Kronemann and Hafner (2010), we have designed a test-bed which provides stigmergic communication to the robots as shown in Figure 6.

In this setting the floor is covered by a glow-in-the-dark foil (i.e. a foil covered by phosphorescent material which absorbs UV light and re-emits the absorbed light at a lower intensity for up to several minutes after the original excitation), and robots are equipped with UV-LEDs pointing toward the floor. Therefore, as robots move around they leave glowing trails behind themselves. Furthermore, for detection of these trails, in contrast to the simple method used by Kronemann and Hafner (2010), in which photo-sensors were used to detect glowing trails, we take advantage of the e-puck on-board camera. By capturing an image and applying a green filter to it, we extract the exact pattern of green trails in the image. The patterns in the image can be used to measure the presence of trail and also its density over different locations. Finally, the IR sensors are used for obstacle avoidance[7].

Experiments

In this subsection we describe the experiments on robotic swarms with limited resources, for two different scenarios of bee foraging and environment coverage, highlighting the benefits of using these type of swarms and practical approaches to overcome their limitations.

Bee Foraging The bee foraging experiments show the effectiveness of the embodied foraging behaviour in a swarm of e-pucks. In Figure 7, we present the stages that the experiment goes through. The goal of the experiment is to show that each separate behaviour actually works in an embodied swarm. Therefore, the experiment starts with a swarm of e-pucks surrounding the hive, see Figure 7(a). Figure 7(b) shows the stage in which a portion of the swarm starts foraging while others remain around the hive, waiting for information to exploit. Figure 7(c) presents the situation in which an exploring e-puck finds food and returns to the hive by using its constructed PI vector. Once returned to the nest, the e-puck communicates its PI findings by means of a virtual dance. The hive collects these experiences and offers these to recruits. Finally, Figure 7(d) gives the situation in which other e-pucks communicated with the hive and have attained the PI vector towards the food source and are traveling to the food source guided by this PI vector. A demonstration movie can be found online[8].

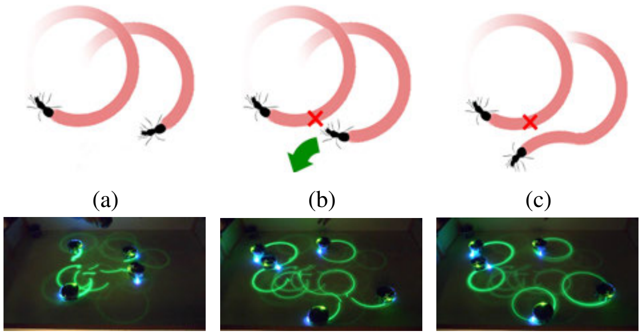
Environment Coverage The multi-robot coverage experiment can be used for various monitoring, rescue, and patrolling missions. Ranjbar-Sahraei et al. (2012b) proposed an stigmergic coverage approach called StiCo which does not need a priori knowledge of the environment, communication among robots or distance measurements. StiCo works based on a very simple motion policy: Each robot circles around with a fixed radius and marks its path with evaporable markers, which denote the borders of robot's territory. Simultaneously, if a robot detects a trail while circling around, it changes its circling direction immediately. This behavior is illustrated in Figures 8(a)-8(c). Ranjbar-Sahraei et al. (2012a) used computer simulations to show that StiCo is very simple, but efficient, robust and even extendable. An illustration of the StiCo coverage approach with real robots is shown in Figures 8(d)-8(f) (Ranjbar-Sahraei et al., 2013b).

Discussion

In summary, the robotic swarms with limited resources can accomplish many tasks successfully. These swarms can use their simple cameras to detect environmental features (shown in Figure 3) and they can communicate using IR

[7] More technical details available in `http://swarmlab.unimaas.nl/stico/indoor-experiments`.

[8] `http://swarmlab.unimaas.nl/papers/bnaic2011demo/`



(a) (b) (c)

(d) (e) (f)

Figure 8: StiCo coordination principle: (a) robots circle around. (b) the right robot detects pheromone. (c) the right robot changes circling direction. (d)-(f) Vision-based Stigmergic Coverage using glowing trails.

communication or XBees (shown in Figure 5). Their small size makes it possible to do experiments in very compact environments (e.g. the testbed shown in Figure 6).

Robotic swarms with limited resources have up and downsides. The upside of these kind of robots is that they are very robust by using lots of proven technologies (e.g., microcontrollers and basic sensors and actuators). They are also relatively cheap to make which for a large swarm could be a more realistic scenario. The downside of using limited resources is that the possibilities of the robotic system is constraint by the hardware platform and the computational possibilities. Also the use of relatively basic sensors limits these kind of platforms in perceiving the environment and interacting or communicating with it. It is also hard to extend the platform with new sensors, which would require some electrical engineering.

We experienced that there are some difficulties, in doing intensive experiments with these swarms: These robots don't contain a modular structure such that in terms of damage, one can replace the damaged part with a new one. Besides, the programmers should usually code all the requirements (e.g, image processing modules) which makes it very time consuming to implement all the requirements and hard to debug.

Concluding Remarks

In this paper we investigated two fundamental problems in swarm robotics, the Foraging and Coverage, from a multi-robot coordination perspective. For the former problem a bee-inspired solution was introduced while pheromone-based communication was used to address the latter problem. For implementation of such problems on real world robotic swarms first a robotic platform with extended resources, Turtlebot, was introduced and the practical requirements to implement the foraging algorithm on a swarm of these robots were discussed. Afterwards, a robotic platform with limited resources, e-puck, was introduced. It was shown how limitations of these kind of robots such as limitation in computational power and low quality vision can be overcome; possible extensions for direct robot-robot communication and the indirect stigmergic communication were considered.

Although the main disadvantage for robots with extended resources is that they are still more expensive and bigger in size and the main disadvantage of the robots with limited resources is that their possibilities are too limited. One can argue that in the near future these argumentations are not relevant anymore. From the current development of System on a Chip (SoC) controllers, used for mobile phones and tablets, one can already see that these low power processors increase processing strength every year, are not really costly and are widely available. Also several sensors and RGBD cameras are miniaturised and will eventually turn up inside a tablet or smartphone. These sensors and controllers will suit the needs of real swarm robotic applications and validate the current use of robots with both limited and extended resources within the field of swarm robotic research.

Considering the pros and cons of each type of studied robotic swarm, one can think of heterogeneous swarms that employ both type of robots in the same mission. This is already explored by the authors in (Ranjbar-Sahraei et al., 2013a) and is still one of the main directions for their future works.

References

Alers, S., Bloembergen, D., Hennes, D., de Jong, S., Kaisers, M., Lemmens, N., Tuyls, K., and Weiss, G. (2011). Bee-inspired foraging in an embodied swarm. In *Proceedings of the Tenth International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1311–1312.

Alers, S., Claes, D., Tuyls, K., and Weiss, G. (2014). Biologically inspired multi-robot foraging. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems (AAMAS)*, page (to appear). International Foundation for Autonomous Agents and Multiagent Systems.

Alers, S. and Hu, J. (2009). Admoveo: A robotic platform for teaching creative programming to designers. In *Proceedings of the 4th International Conference on E-Learning and Games: Learning by Playing. Game-based Education System Design and Development (Edutainment)*, Edutainment '09, pages 410–421, Berlin, Heidelberg. Springer-Verlag.

Alers, S., Ranjbar-Sahraei, B., May, S., Tuyls, K., and Weiss, G. (2013a). Evaluation of an experimental framework for exploiting vision in swarm robotics. In *Advances in Artificial Life, ECAL*, volume 12, pages 775–782.

Alers, S., Ranjbar-Sahraei, B., May, S., Tuyls, K., and Weiss, G. (2013b). An experimental framework for exploiting vision in swarm robotics. In *ADAPTIVE 2013, The Fifth International Conference on Adaptive and Self-Adaptive Systems and Applications*, pages 83–88.

Blum, C. and Sampels, M. (2004). An ant colony optimization algorithm for shop scheduling problems. *Journal of Mathematical Modelling and Algorithms*, 3(3):285–308.

Breitenmoser, A., Schwager, M., Metzger, J., Siegwart, R., and Rus, D. (2010). Voronoi coverage of non-convex environments with a group of networked robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4982–4989.

Cheng, K., Collett, T., Pickhard, A., and Wehner, R. (1987). The use of visual landmarks by honeybees: Bees weight landmarks according to their distance from the goal. *Journal of Comparative Physiology A*, 161(3):469–475.

Claes, D., Hennes, D., Tuyls, K., and Meeussen, W. (2012). Collision avoidance under bounded localization uncertainty. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2012)*, Vilamoura, Portugal.

Collett, M., Collett, T. S., Bisch, S., and Wehner, R. (1998). Local and global vectors in desert ant navigation. *Nature*, 394(6690):269–272.

Collett, M., Harland, D., and Collett, T. S. (2002). The use of landmarks and panoramic context in the performance of local vectors by navigating honeybees. *The Journal of Experimental Biology*, 205:807–814.

Di Caro, G., Ducatelle, F., and Gambardella, L. M. (2005). Swarm intelligence for routing in mobile ad hoc networks. In *Proceedings of the IEEE Swarm Intelligence Symposium*, pages 76–83.

Dorigo, M., Birattari, M., and Stutzle, T. (2006). Ant colony optimization: Artificial ants as a computational intelligence technique. *Computational Intelligence Magazine, IEEE*, 1(4):28–39.

Dorigo, M. and Blumb, C. (2005). Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344:243–278.

Dorigo, M. and Stützle, T. (2004). *Ant Colony Optimization*. A Bradford book. BRADFORD BOOK.

Dressler, F. and Akan, O. B. (2010). A survey on bio-inspired networking. *Computer Networks*, 54(6):881–900.

Floreano, D. and Mattiussi, C. (2008). *Bio-inspired artificial intelligence: theories, methods, and technologies*. The MIT Press.

Fujisawa, R., Imamura, H., Hashimoto, T., and Matsuno, F. (2008). Communication using pheromone field for multiple robots. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 1391 –1396.

Herianto, Sakakibara, T., and Kurabayashi, D. (2007). Artificial pheromone system using rfid for navigation of autonomous robots. *Journal of Bionic Engineering*, 4(4):245 – 253.

Johansson, R. and Saffiotti, A. (2009). Navigating by stigmergy: A realization on an rfid floor for minimalistic robots. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 245 –252.

Kronemann, M. L. and Hafner, V. V. (2010). Lumibots - making emergence graspable in a swarm of robots. In *The ACM Designing Interactive Systems Conference*, pages 408–411.

Lemmens, N. (2011). *Bee-inspired Distributed Optimization*. Maastricht University.

Lemmens, N., Alers, S., and Tuyls, K. (2011). Bee-inspired foraging in a real-life autonomous robot collective. In *Proceedings of the 23rd Benelux Conference on Artificial Intelligence (BNAIC)*, pages 459–460.

Lemmens, N. and Tuyls, K. (2012). Stigmergic landmark optimization. *Advances in Complex Systems*, 15(8).

Mondada, F., Bonani, M., et al. (2009). The e-puck, a robot designed for education in engineering. In *9th Conference on Autonomous Robot Systems and Competitions*, volume 1, pages 59–65. IPCB: Instituto Politcnico de Castelo Branco.

Ranjbar-Sahraei, B., Alers, S., Stankova, K., Tuyl, K., and Weiss., G. (2013a). Towards soft heterogeneity in robotic swarms. In *Proceedings of the 25th Benelux Conference on Artificial Intelligence (BNAIC)*, pages 384–385.

Ranjbar-Sahraei, B., Alers, S., Tuyls, K., and Weiss, G. (2013b). Stico in action. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 1403–1404. International Foundation for Autonomous Agents and Multiagent Systems.

Ranjbar-Sahraei, B., Weiss, G., and Nakisaee, A. (2012a). A multi-robot coverage approach based on stigmergic communication. In *Multiagent System Technologies*, volume 7598 of *Lecture Notes in Computer Science*, pages 126–138. Springer.

Ranjbar-Sahraei, B., Weiss, G., and Nakisaee, A. (2012b). Stigmergic coverage algorithm for multi-robot systems (demonstration). In *Proceedings of the Eleventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, volume 3, pages 1497–1498.

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

Designing a Robotic Platform Controlled by Cultured Neural Cells

Norihiro Maruyama[1*], Atsushi Masumori[1*], Julien Hubert[1], Takeshi Mita[1],
Douglas Bakkum[2], Hirokazu Takahashi[1] and Takashi Ikegami[1]

[1]The University of Tokyo
[2]ETH Zurich
[*]These authors contributed equally to this work.
maruyama@sacral.c.u-tokyo.ac.jp

Abstract

Robot experiments using real cultured neural cells as controllers are a way to explore the idea of embodied cognition. Real cultured neural cells have innate plasticity and a sensory motor coupling is expected to develop the neural circuit. We designed a system in which a robot moving in a real environment is controlled by cultured neural cells growing on a glass plate attached to a High-Density Microelectrode CMOS Array(HDMEA). The IR sensors on a robot will feedback onto the neural cells through HDMEA and the activity of the neural cells will be read again by HDMEA and sent back to determine the speed of the robot. Most of the previous works have used the relatively low-density multi-electrode array for recording and stimulating the neural assembly. Our system has the advantage of a high-density spatial and temporal array so that we can precisely detect which neurons get fired and suppressed. A preliminary finding from the experiment is that synchronized neural activation is retained in cultured neurons even after detached from a robot.

Introduction

Recently, it became easier and popular to study the coupling between a robot and a network of cultured neural cells. In those studies, the sensory information coming from the moving robot is used to stimulate the neural cells, and the resulting activities of those determine the speed of the motors driving the robot. This is what is called a "closed loop" experiment. We believe that it is critical to conduct such closed loop experiment for revealing biological memory and adaptability with respect to embodiment. Behavior is not a one way function of sensory inputs but behavior assimilates by itself.

For example, Bakkum et al. (2008) have proposed a new method to train a biological neural cells to achieve a desired pattern for multiple stimulus. Kudoh et al. (2008) have proposed another learning method using a cultured neural system that incrementally learns to respond in a particular way to a particular input. One drawbacks of those studies is that their microelectrode array has not enough space resolution, so that it is difficult to stimulate/detect a single neuronal state. The other drawback is giving an external evaluation function that enables a coupled neuro-robot system to work. Such evaluation function should be developed from the neuro-robot itself, namely, we have to develop neural self-organization of sense-making behavior with a mobile body. In order to overcome those drawbacks, we use a recently developed high density CMOS array (HD-MEA) capable of detecting the activity of individual neurons with high precision. With HMDEA, we can measure the spatio-temporal neural pattern with a higher precision and reveal how neural plasticity and memory can self-organize the sense-making behavior in a given environment.

Method

A simplest task we seek here is avoiding or reaching behavior of the robot without putting further constraints.

The main components of this system are the HDMEA monitoring the culture of neural cells, the robot in its arena and the interface connecting them. The system gathers the signal from the robot, stimulates the neural cells by using HMDEA and sends the motor output signal to the robot. This way, the robot and the neural cells form a closed loop. The HDMEA we used in this study provides a higher spatio-temporal resolution compared to previous studies [Frey et al. (2010)]. Thus we can monitor all the neural activities by using less than 126 cultured neural cells and the adequate electrode channels. For the moment, we can stimulate at most two cells at a time out of 126 channels available. The sampling rate of HDMEA is 20kHz. A software MeaBench developed by D. Wagenaar [Wagenaar et al. (2005)] is used for recording and detecting spikes and controlling the whole system.

We used Elisa-3 (Manufactured by CGtronic) as a mobile robot. Elisa-3 is a circular small robot of 2.5 cm radius and it has two independently controllable wheels. In this experiment, we use the front right and front left distance sensors as sensory stimulation for the neural cells.

We chose two excitatory neurons as left or right input-neuron for receiving the stimuli. Stimulation to the neural cells is determined by the in-take sensor inputs of this robot. We designed it as that the closer a robot approaches a wall and the higher the sensory inputs become, the more