Re-Embodiment of Honeybee Aggregation Behavior
in an Artificial Micro-Robotic System

S. Kornienko¹, R. Thenius², O. Kornienko³, T. Schmickl⁴

¹,³ Institute of Parallel and Distributed Systems, University of Stuttgart,
Universitätsstr. 38, D-70569 Stuttgart, Germany
{korniesi, kornieoa}@ipvs.uni-stuttgart.de

²,⁴ Department for Zoology, University of Graz,
Universitätsplatz 2, A-8010 Graz, Austria
theniusr@gmx.de, schmickl@nextra.at

Contact author: S. Kornienko, phone: +49-711-7816373, fax: +49-711-7816250
Abstract

In this paper we describe the re-embodiment of biological aggregation behavior of honeybees in Jasmine micro robots. The observed insect behavior, in the context of the insect’s sensor-actor system, is formalized as behavioral and motion-sensing meta-models. These meta-models are transformed into a sensor-actor system of micro-robots by means of a sensors virtualization technique. This allows us to keep the efficiency and scalability of the bio-inspired approach. We also demonstrate the systematic character of this re-embodiment procedure on collective aggregation in real robotic swarm.

Keywords: Biologically-inspired algorithms, micro-robotic swarm, aggregation behavior, embodiment, sensor-actor couplings.

Shortened title: Re-Embodiment in a Micro-Robotic System.
Re-Embodiment of Honeybee Aggregation Behavior in an Artificial Micro-Robotic System

Abstract

In this paper we describe the re-embodiment of biological aggregation behavior of honeybees in Jasmine micro robots. The observed insect behavior, in the context of the insect’s sensor-actor system, is formalized as behavioral and motion-sensing meta-models. These meta-models are transformed into a sensor-actor system of micro-robots by means of a sensors virtualization technique. This allows us to keep the efficiency and scalability of the bio-inspired approach. We also demonstrate the systematic character of this re-embodiment procedure on collective aggregation in real robotic swarm.

1 Introduction

"Swarm intelligence" is a phenomenon found in social insects (Bonabeau et al., 1999). This feature of distributed self-organized processes is characterized by the ability of a group of animals to solve a common problem collectively. Usually, a single worker animal is not capable of solving a problem alone, but solving it collectively produces an increase in the collective fitness of the colony. Examples for swarm intelligence are e.g. the selection of foraging targets performed by ants and honey bees; the self-organized nest construction of ants, bees, termites and wasps; sorting tasks.

Biological strategies of collective problem-solving can successfully be used in artificial micro-robotic systems (Sahin, 2004). Micro-robots cannot act individually in solving problems (I-Swarm, 2003-2007), due to their limitations in size, available on-board energy, actuation, sensors, and communication capabilities. Technically useful behavior can be achieved when robots not only coordinate their behavior but also
when a problem solving strategy is designed for a collective solution (Kornienko et al., 2004). Swarm-robotic experiments clearly demonstrated, that increasing the number of robots that work on one problem in parallel, does not lead to linear or sub-linear improvement of collective performance, e.g. (Jimenez, 2005). Moreover, the growth of a robot population over some threshold decreases collective fitness because of overhead and deadlock in communication and coordination. Simple parallel solutions do not increase the fitness of a robot swarm.

Biological solutions possess structures that are required by the most problem-solving approaches (Camazine et al., 2003), due to their long-term natural optimization via evolution. The key issues are efficient and scalable coordination mechanisms, specific incorporation of global and local information sources as control signals, and full independence of individual behavior, where even a large number of agents does not hinder themselves. Unfortunately, these highly-optimized natural solutions cannot directly be applied to robotic systems. The point is that the individual behavior of each biological agent reflects its own sensor and actuator system. These systems are too different in insects and micro-robots to be directly transferred from one entity to the other. Indeed, we believe the "silicon-metal" robot cannot, even in the near future, mimic the sensor and actor capabilities of real biological organisms.

To use advantages of biological strategies for collective robotic systems, these strategies have to be re-embodied for another sensor-actor system (Kornienko et al., 2005b). Here we encounter the challenge of keeping the efficiency of biologically-inspired approaches despite their different final implementation. In this work we demonstrate not only this particular re-embodiment procedure, but also its systematic nature by means of a "virtualization" technique. This is an important topic for the swarm-robotic and adaptive-behavior research community as an adaptation approach of biologically-inspired mechanisms for real large-scale swarms.

The rest of the paper is organized as follows. The experiments with honeybees thermotactic aggregation behavior is described in Section 2. Re-embodiment of bio-inspired strategies for robotic system is presented in Section 3 and the experiments in Section 4.
Figure 1: A group of 32 bees aggregating on a flat gradient. The bees were released in the colder left part of the arena (approx. 25 °C). After some time, the bees aggregate on the right side of the arena, which was 36 °C. During the runs, bees formed clusters in colder areas of the arena, but most of bees moved to the optimal place.

2 Honeybees thermotactic aggregation behavior

As the test scenario we used a thermotactic aggregation behavior (Heran, 1952) (Grodzicki & Caputa, 2005) (Crailsheim et al., 1999) found in honeybees. On flat temperature gradients, complex collective behavior is observed. This behavior is expressed by the formation and disaggregation of clusters throughout the temperature field. After some time only one big cluster emerges, which is located at (or near) the region of optimal temperature (see Figure 1). This collective behavior shows some analogies to other collective behaviors found in the animals: clustering and aggregating of animals via positive feedback, "sorting problems".

On the one hand, there is "item sorting", found in ants, termites and bees (Theraulaz & Bonabeau, 1995). In this case, items like "food" or "brood" are transported by the workers from one place to another and aggregated in several distinct places. Often the workers aggregate items of different kinds in different places, thus they per-
form qualitative sorting. In this way, seeds can be sorted by size or by kind, or brood stages can be sorted, so that eggs, larvae and pupae are separated. The process of sorting and aggregating items is purely self-organized, but often it is also modulated by environmental conditions (humidity, temperature, distance to queen etc.). If the aggregated items are building blocks of the nest (sand, mud etc.), then the same process is used for nest-construction. This is found in the dynamic shaping of the queens chamber in termites and in the radial nest structure of some ant species (Bonabeau et al., 1999), (Camazine et al., 2003).

On the other hand, the animals often aggregate themselves ("animal sorting") in specific areas. The most prominent cases are the reproductive swarms and the winter-clusters of honeybees (Myerscough, 1993) (Sumpter & Broomhead, 2000), (Watmough & Camazine, 1995). Other examples include aggregation of ants in the nest (Theraulaz et al., 2002), chain formation in ants (Lioni et al., 2001), the aggregation of cockroaches (Jost et al., 2004), (Jeanson et al., 2005), of bark beetle (Camazine et al., 2003), and of slime-mold amoebas and of several bacteria species (Camazine et al., 2003). All these phenomena work via simple positive feedback loops and they all have in common that aggregation is performed without any central regulatory unit. These examples provide bio-inspiration for technical approaches to aggregation scenarios in real robot swarms.

In this article we propose a biologically inspired aggregation algorithm for mobile micro-robots based on the thermotactic behavior found in honeybees. Aggregation of honeybees is strongly dependent on the form of the temperature gradient. The bees find the point of optimal (high) temperature, based on the difference between the warmest and the coldest point in the arena. If the gradient is not very steep, the bees are not able to determine the uphill-direction in the gradient, and perform mainly a random walk. Nevertheless, they are still able to find the place of the optimal temperature, but they can only find it collectively.

2.1 Analysis of the bees behavior

We performed experiments to investigate the aggregation behavior of young honeybees. A cohort of young (1 day old) bees was introduced into an arena consisting
of a comb plate (floor) and 4 walls. The apparatus was set up in a dark room. A temperature gradient was established using a IR-lamp (pulp) that was shielded by a SCHOTT \textsuperscript{tm} filter. Such a filter allows only red and IR light to pass through. These wavelengths are invisible to honeybees, so they were navigating only in the temperature field and had no visual cues. An IR-sensitive camera was mounted on top of the arena. We report only selected data on the behavioral analysis of the bees, because our goal is to demonstrate the re-embodiment of the bees’ individual behavior. The full analysis of the global patterns and of individual behavior will be published separately.

A representative pattern emerging in the temperature arena is depicted in Figure 1. For the following analysis, we defined 4 concentric zones that were arranged around the spot of the optimal temperature. The arrangement of these zones is depicted in Figures 2 and 3. Zone 1 was located directly below (and around) the lamp, it had the highest average temperature (34.5±1.5°C). Zone 4 was located at the remote end of the arena and had the lowest average temperature, which was 25.5±0.5°C.

![Figure 2: Measurements of the temperature field in the bee arena.](image-url)
The analysis of the emerging aggregation pattern showed that more clusters are formed in those zones that are near the optimum point. Near the optimum place in the arena, larger clusters are formed and clusters survived longer, compared to remote areas of the arena (see Figure 4).

We also analyzed individual behavior of bees. We observed 9 randomly selected bees for 10 minutes and recorded all behaviors (walking, resting) and all events (collisions) that happened during the observation using Noldus™ Observer.

For the re-embodiment of the honeybee behavior, the first question we had to answer was: Are the bees able to discriminate between a collision with a wall and a collision with another bee? To analyze this we browsed the observation-logs for all collision events and classified these events into four categories: collision with the wall and stop of motion afterward, collision with the wall and moving on, collision with another bee and stop of motion and collision with another bee and moving on.

As illustrated in Figure 5, the bees were able to discriminate between other bees and walls, because they almost always stopped next to other bees: when a bee collided with a wall, it did not stop there.

The final question was: Why are clusters at lower temperature (far away from the optimum) shorter than clusters near the optimum? Our assumption was, that bees have two variables that they optimize: social contact and temperature. We also assumed that the resting time of a bee in a cluster depends on local temperature.
Figure 4: Clustering behavior of focal bees (N=9 bees). We recorded the fraction of observation time that was spent in a cluster by each focal bee. These data were classified into the 4 temperature zones and by social contact, that is the size of the cluster the bees were located in.

Analysis showed, that in the warmer zones, the mean resting time of bees was indeed longer in the warmer zones than in the colder zones (see Figure 6). Based on these findings, we can summarize the core behavioral algorithm of an individual bee as depicted in Figure 7.

We assumed that the reason for the non-directed random walk of bees in our temperature field was because a flat gradient could not be exploited by a single bee when moving uphill in the gradient. Perhaps the temperature difference was too small at the sensors of the bee, which are located mainly in their antennae. The bees move almost randomly and often stop when they hit another bee. In this way a cluster is formed and by chance (depending on the density of bees) other bees can join the cluster. The bigger the cluster is, the more likely free-running bees are captured by the cluster and the more unlikely it is that the cluster will decrease to a size of one. In addition the probability that bees leave a cluster depends on the local temperature and not on the local gradient. The warmer it is, the longer the bee stays in the cluster. Thus, in the initial phase, many clusters are formed, but over time,
sub-optimal clusters shrink and near-optimal clusters grow. Finally, only one cluster remains at the optimal position in the arena.

Next we implement a distributed algorithm in a swarm of mobile robots by using a light gradient instead of a temperature gradient as environmental template. We assumed that the resulting collective behavior of the robot-swarm would show the following properties, according to the principles mentioned in (Kennedy & Eberhart, 2001), (Millonas, 1994).

**Stability.** The swarm finds a stable final solution. The solution does not depend on the initial distribution of robots in the arena. In addition, the measurement of the sensory data is only performed when the robots are standing in clusters. In cases where measurement is very noisy\(^1\), the measurement is performed many times and then averaged. In the hypothesized algorithm, no sensor data (except collision data)

\(^1\)For example in the extreme parts of the sensor range: at very low illumination or at very high illumination when sensors are almost saturated.
Figure 6: Median and inter-quartile ranges of observed resting times of bees in the four defined temperature zones. $N = 9$ observed bees.

is used during the moving phase of the robot.

**Flexibility.** The collective decision of the swarm is flexible, so after a translation of the place of optimal light, the robots will collectively find the new optimum.

**Scaling properties.** The intelligence does not reside within one single individual. One individual alone cannot find a solution using this algorithm. The more individuals that perform this behavior simultaneously, the faster and the more precise the optimum solution is found. The suggested algorithm works better with higher number of bees than with lower number of bees. An algorithm that depends more on comparisons performed by individuals (like a greedy uphill walker), will work most efficient if performed by one agent alone. A collective based algorithm should work better if it is performed by many agents in parallel.

**Computational effort.** Other algorithms will require much more computational
effort and sensory abilities within each single agent. For example, three possible simple alternative algorithms for the same problem are discussed (Mletzko, 2006), that all work purely within each single robot and do not involve social interaction among the robots:

1) Greedy uphill-walker: needs two sensors and a periodic assessment of two temperature sensors. In addition, a control mechanism is needed that steers the robot in the appropriate direction.

2) A robot that has just one sensor and exploits the gradient: this robot requires a memory to store at least the last measurement and to compare it with the current one. Afterward, a computation of the gradient with respect to the travelled distance can be performed to evaluate the gradient. Again, a sort of steering-algorithm to drive the robot uphill is also needed.

3) A robot that moves randomly and stops if the desired temperature is achieved. This robot would still need to evaluate the environment periodically and will need memory.

In contrast to our suggested bio-inspired algorithm, all three algorithms, described above, would need efficient collision avoidance additionally if they are performed with
several robots in parallel. The algorithm we propose achieves the same goal, without the need for permanent updating of light-sensor data, without an uphill-steering algorithm, without memory and without the need for a second sensor. The hypothesized algorithm minimizes sensor and computation, because the local light sensor data are evaluated only once per collision with another robot. This is the reason why the performance of the swarm increases with increasing density of robots in the arena: more frequent collisions lead to a higher update frequency of sensor data during the initial phase and the environment is collectively scanned by the swarm in smaller area-intervals. After the initial clusters are formed, the collision frequency decreases automatically and so also the computational effort is minimized. After environmental fluctuation, the robots move around again, as the previous cluster dissolves and a new one forms, thus requiring less computational effort.

3 Re-embodiment of individual behavior for micro-robot

3.1 Description of the micro-robot Jasmine

For performing the swarm-experiments and testing the embodiment concept we used the micro-robots ”Jasmine”, see Figure 8. It is a public open-hardware development at www.swarmrobot.org, having a goal of creating a simple and cost-effective micro-robotic platform and knowledge exchange in the swarm robotics community.

The micro-robot is $26 \times 26 \times 20$mm, uses the two Atmel AVR Mega micro-controllers: Atmel Mega88 (motor control, odometry, touch, color and internal energy sensing) and Mega168 (communication, sensing, perception, remote control and user defined tasks). Both micro-controllers communicate through high-speed two-wired TWI (I2C) interface. It has on board 24kb flash memory for program code, 2kb RAM for data and 1kb nonvolatile memory for saving working data.

The robot has six ($60^\circ$ opening angle) communication channels (also used for proximity sensing) and one geometry-perception-channel ($15^\circ$ opening angle) based on separate IR receivers and transmitters. Communication area covers $360^\circ$ rose-like-areal with maximal and minimal ranges of 200mm and 100mm respectively. The physical communication range can be decreased through a change of sub-modulation.
The robot also has a remote control and robot-host communication (up-link and down-link), which is isolated from all other channels (through modulation).

The robot uses two DC motors with internal gears, two differentially driven wheels on one axis with a geared motor-wheels coupling. Encoder-less odometrical system normalizes a motion of the robot (the robot is able to move straight forwards and backwards), and estimates the distance travelled with accuracy of about 6% and rotation angle - of 11%. Jasmine III uses 3V power supply (from 3.7V Li-Po accumulator) with internal IC-stabilization of voltage. Power consumption during motion is about 200mA, in stand - 6mA, in stand-by mode less 1 mA. Power lasts from 1 to 2 hours during autonomous work. The robot is also capable of autonomous docking and recharging, so that a real time of experiments is effectively unlimited.

Robots are programmed using C with open-source gcc compiler. There is a complete BIOS system that supports all low-level functions. Moreover, for quick implementation of swarm behavior there is a jasmine-SDK system that includes an operational system and high-level functions based on Petri nets, see Figure 9.

Basically, the operational system executes four steps repeatedly: read of sensor
Figure 9: (a) Structure of jasmine-SDK; (b) Operational system of the robot.
data, perform communication, make decisions and finally execute a plan. The interruption service takes care of software and hardware interruptions. The plan, that a robot has to execute, represents a Petri net consisting of two parts: service part (handlers for interruptions) and user-defined part (behavioral program for the robot). The structure of the service part represents an interruption vectors system with corresponding handlers. The interruptions (like touch or low-energy) are generated by BIOS system, users only need to write the corresponding handlers. For further details of construction and programming see swarmrobot.org or e.g. (Kornienko et al., 2005a).

3.2 Re-embodiment of biological strategy

For re-embodiment we used a 4-step methodological approach, sketched in Figure 10. Firstly, we try to formalize the sensor/actor capabilities of natural agents (bees), especially those, which are assumed to be relevant for aggregation. Then the observed aggregation behavior of these agents is expressed in terms of their sensor-actor capa-

![Figure 10: The structure of re-embodiment procedure.](image-url)
Re-Embodiment in a Micro-Robotic System

Capabilities, in other words, expressed as meta-models. Even in this early step we have to take into account realistic capabilities of artificial agents.

In the next step, the sensor/actor capabilities of natural agents are mapped into available sensor/actor capabilities of artificial agents. Here we have to take into account technological restrictions, such as characteristics and functionality of sensors, degrees of freedom (DOF) and holonomicity of motion system, available energy and so on. This step represents a ”virtualization” of sensing and actuation. For example, when some required sensor capabilities cannot directly be reproduced, they can be indirectly obtained by other sensors and ”adapted” to the required ones by software.

The last step represents a trade-off between ”virtualized” sensor/actor capabilities of artificial agents and the original behavioral strategy of natural agents. Obviously, not all required sensors and actors can even be ”virtually” implemented, so that an adaptation of original strategy is required. However this adaptation should be performed carefully so as to keep the structure of bio-inspired algorithm, because this structure is expected to fit best the requirements of collective systems.

After the original behavioral strategy is implemented, experiments can be performed. In some cases (it happened in our re-embodiments experiment) the ”virtualized” sensors/actors are still not suitable. In this case, a new implementation and ”virtualization” of sensors and actuators should be performed until the emerged behavior becomes similar to the behavior exhibited by natural agents.

1. Representation of behavioral rules of natural agents in terms of their sensor/actor capabilities.

For re-implementation of biological strategies, we first tried to understand the main differences in observable behavior of bees and known capabilities of micro-robots, shown in Table 1. As shown in this Table, there were several principal sensor/actor issues that made biological aggregation possible:

- Detection of temperature changes;
- Directional sensing of these changes (to detect a gradient);
- Detection of other bees and their differentiation from other objects;
Bee behavior, sensor-actor system | Robot capabilities
--- | ---
Bees are able to differentiate between other bees and obstacles. This is achieved by cuticula-bound chemical substances (taste, smell). | This differentiation is achieved via emission of light signals by robots, which mimic the chemical substances on the bees surfaces.
The duration of resting time in a cluster is a function of the local temperature. | The duration of resting time in a cluster is a function of the local illumination.
Perception radius - bees detect other bees at close range or only by touch. | A robot can detect another robot by detecting its emitted IR-radiation. A robot cannot detect a contact with its chassis but this can be simulated by proximity sensing and rotation.
In flat gradients, the detection of the direction toward the temperature optimum seems not to be possible for a bee - it performs almost a random walk. | In robots, we do not measure the illumination during the movement phase. Our robots move straight and turn randomly after a collision with the border of the arena.
Bees are able to build a dense cluster with a high degree of connectivity. | Robots cannot build a dense cluster, because of the 2 DOF motion and a need of a space for rotation.
Bees walk relatively slow, 1-2 body/s and the motion system can be described as holonomic. | A robot can change its velocity in a wide range, however, experiments should be done in low-velocity mode. The motion is non-holonomic.

Table 1: *Comparison between observable behavior of bees and known capabilities of micro-robots.*

- Behavior in dense clusters with a high degree of connectivity.

The idea of biological aggregation in terms of bees sensor/actor system is the following: when the bees meet in a region with higher temperature than ambient, they walk more slowly. The more bees that join a cluster, the more bees in this cluster are "blocked" by other bees and therefore the longer such a cluster can persist. The cluster is then available longer for other bees to join, which represents a positive feedback loop. In this way, the aggregation is a specific relationship between sensitivity of sensors and holonomicity of motion.

There is a topic assumed to have an essential impact on the global aggregation patterns: the building of seed points. Such "seed points" are initial aggregations of two bees, which allow a further growing of clusters. The two-bees clusters do not exist a long time because bees are not "blocked" within them. Therefore we assume the
existence of "a seed point factor": either the bees density for cluster formation should be much higher than in a normal case, or bees have a mechanism allowing two bees to stay together for a long time.

2. "Virtualization" of sensor/actor system in artificial agents:

Replication of the biological aggregation is not directly possible because of the following reasons:

1. The robots have no ability to sense temperature.

2. Due to their sensor-actor capabilities, these micro-robots cannot achieve clusters with high densities, this fact can essentially change the behavior during a "blocking phase" and finally lead to a cluster disaggregation.

These shortcomings of the robots can be compensated by the following "virtualization" techniques.

I. Instead of temperature, we used a light gradient that approximatively resembles the temperature gradient that was used in the bee experiments. However the light intensity differs in several ways from temperature: light in not cumulative and the light intensity has different distribution on the surface than does temperature. This point is demonstrated in Figure 11(a) in the preliminary experiments. To allow a few robots to aggregate, the light region should be large enough and should not contain IR-spectra. This is why we used a special round luminescent lamp for generation of the light gradient. The light intensity in the brightest region (region A) is almost constant within the whole region A. The highest variance in intensity is found in the transient region (region B), whereas the rest of surface has again small changes in intensity (region C). In this way, a light gradient can be established only in the small region B.

For re-embodiment we need sensors that can detect a light intensity with sufficient sensitivity without being saturated (to work equally linearly in bright and dark regions). For that we first used small 3×3mm clear-epoxy encapsulated solar cells, generating about 1/2 volt, 1.9mA in sunlight, see Figure 12(a). They are connected
directly to the ADC port of the microcontroller. To implement a gradient-search strategy using one light sensor, the robot can rotate in the radius of approximately 5-7cm and save the values of light intensity. However, after performing the preliminary experiments we arrived at the conclusion that this sensor, despite the linearity of the light-signal transformation, cannot be used in aggregation experiments. The reason was due to low sensitivity to light changes, especially in dark regions. For example, the robot was not able to identify the region B, shown in Figure 11(a). Therefore, after the preliminary experiments, we decided to change the light sensors to the more sensitive type APDS 9002, produced by Agilent Technologies, and installed two of them for further experiments with gradient light on a small extension board (Figure 12(b)).

The comparison of light-signal conversion for both sensors is shown in Figure 12(c). The solar cell is much more linear and inertial for the light changes, whereas APDS 9002 is more sensitive and has essentially less reaction time. This sensor is able to detect light changes of AC luminescent lamp, so we installed a passive RC-filter on the extension board and averaged the obtained values (dispersion of APDS 9002 is much higher than the solar cell as visible in Figure 12(c)).

II. To implement the virtual sensor, capable of distinguishing between objects and other robots, we used the following idea. The robot emits the IR light for proximity
sensing and percept the IR-light reflected from the obstacles in 6 channels (see Figure 12(a)). This emitted IR-light can be sensed by a robot only at a short distance and can be interpreted as a “contact pheromone”, see Figure 11(b). When a robot detect IR-signals, it assumes that some robots are in its local neighborhood. The robot then emits identifiable by other robots. This strategy underlies the “virtualized robot-recognition” sensor, its algorithm is shown in Figure 13. In this way, by a combination of existing sensors, specific behavior and signal processing we are able to make a ”virtualized” sensor of required functionality.

3. Adaptation of original behavioral rules to ”virtualized” sensor/actor capabilities of artificial agents.

The behavioral pattern, shown in Figure 7, can be adapted to other motion capabilities of the robot (in the cluster building part, especially seed points) and to other sensing capabilities (primarily parameters of waiting).

(I) For adaptation of the algorithm for establishing a seed point in robotic experiments, the robots have to meet and to stay long enough together so that other robots can join the cluster. We can analytically calculate this relation by taking several analogies to molecular-kinetic theory of ideal gas, such as a diffusion in ideal gas. We introduce the following notions: the sensing radius $R$ ($R_s$, $R_c$ are collision avoiding and communication radiiuses), $l_c$ the length of free path from the start of motion till the first communication contact; $l_s$ the length of free path from the start of motion till the first collision-avoiding contact; $n_c$ and $n_s$ are correspondingly the number of communication and collision-avoiding contacts; $S_c$ and $S_s$ are the area of the ”broken” rectangles created by motion in time interval $t$ with $R_c$ and $R_s$, and finally the robots motion velocity $v$. We do not consider collision avoidance caused by obstacles, changes in robots velocity or behavior during collision avoidance. This simplifies calculations and we expect that this approximation error is not large for small clusters. Figure 14 illustrates this idea.

We differentiate between communication contacts and collision avoiding contacts. For further calculation we use collision avoiding contacts, because they correspond
Figure 12: (a) The first version of the light-sensor (solar cell) installed directly on the robot. Shown are also IR-emitters and receivers; (b) The second version of the light sensor (sensor APDS 9002), installed on the robot as an extension board; (c) Sensitivity and nonlinearity of the first and second version sensors in the light-signal transformation. Measured are the light at different distances from the luminescent lamp (as shown in Figure 11 (a)), distance between arena surface and the lamp 12cm, ambient light was off), the values of solar-cell sensor are multiplied by 6. In the signals from APDS 9002 we can easily recognize the corresponding Regions from Figure 11(a).
more to a bio-inspired approach. Firstly, we are interested in the number of two-robots contacts $n_s$ that happen during the movement. This value is equal to the average number of robots in the area $S_s$,

$$n_c = S_s D_{sw},$$

where $D_{sw}$ is the swarm density. We assume that the robot’s rotation radius is small (robots can rotate in one place), so that we can neglect the area of fractures. In this case $S_s = 2R_s vt$. $D_{sw}$ can be calculated as the number of robots $N$ in a swarm divided by the area available for the whole swarm ($S_{sw} - NS_r$):

$$D_{sw} = \frac{N}{S_{sw}} \quad \rightarrow \quad n_s = \frac{2R_s vt N}{S_{sw} - NS_r},$$

where $S_{sw}$ is the whole area, $S_r$ is the area occupied by robot itself. In the relation (2) we assume only one robot moves while the others are motionless. More exact relation for the case when all robots move differs from (2) only by the numeric coefficient $\sqrt{2}$.
Re-Embodiment in a Micro-Robotic System 24

Figure 14: Motion path of a robot with communication and collision-avoiding contacts.

(as proved by Maxwell for a diffusion in ideal gas). For further calculation we use

\[ n_s = \frac{2\sqrt{2}R_svtN}{S_{sw} - NS_r} \rightarrow t = \frac{(S_{sw} - NS_r)n_s}{2\sqrt{2}R_svtN}. \]  

(3)

Obviously, that the two-robot contact \((n_s = 2)\) will happen during the time \(t_2\) and after this, the number of available robots is decreased by one (two robots after contact stop and are considered as one robot of double size). The time for the next contact can be calculated by using (3) when \(N=N-1\), we are interested only in a small number of total contacts (when number of robots in a cluster is smaller than a number of available robots, we can skip changes in behavior caused by non-holonomic motion).

We can express the total time \(t_k\) of \(k\)-contacts \((N/2 > k > 2)\) as

\[ t_k = \frac{S_{sw} - NS_r}{2\sqrt{2}R_svt} \sum_{i=2}^{i=k} \frac{i}{N - i + 2}. \]  

(4)

For calculation, we took \(S_{sw} = 1000 \times 1000mm^2\), \(S_r = 26 \times 26mm^2\), \(R_s = 100mm\), \(v = 50mm/sec\) and \(N = 10\). For two-robot contact we get a typical time for \(t_2 \approx 14sec., t_3 \approx 37sec., t_4 \approx 72sec.\) i.e. robots 23-35 sec. do not move and wait till the next robot comes into the cluster. The dependencies between \(t, Rc\) and \(N\) for different values are shown in Figure 15. These periods agreed with experimental data from the preliminary experiments with micro-robots.

As followed from the calculation, the seed point for building clusters can appear
when robots wait at two-robot contact at least 23-35 sec. (this time can decrease when increasing swarm density). The waiting (resting) time matches very well with the sensor-related adaptation of the algorithm.

(II) For adaptation of the original bee algorithm shown in Figure 7 for other sensors, we measured the light gradient in the robot arena (as we measured the temperature gradient in the honey bees). Figure 16 shows the results of these measurements. The light gradient in the arena (Figure 16) is very similar to the temperature gradient.
in the honeybee arena (Figure 2). The main difference is that the temperature scaled between 25°C and 36°C while the measurable light intensity values were between 0 and 145. For a successful embodiment, we developed a conversion rate between these two measurements (see Equation 5):

\[ temperature_{local} = 0.0759 \times \text{light value} + 25.0. \] (5)

The duration of the bees in the temperature zones (see Figure 6) showed a clear nonlinear effect. In the zone 1, the resting times were the longest. To incorporate this into the robot algorithm, we implemented a non-linear stimulus-response-curve, which is depicted in Figure 17.

![Figure 17: The stimulus-response-curve we used in our robot experiments. The higher the sensory values of the light sensor were, the longer the robot waited in a cluster.](image)

4 Implementation of aggregation strategy for micro-robotic swarm and performing the experiments

The structure of the algorithm for robot behavior is shown in Figure 18. After the
robot encounters an obstacle, the corresponding interruption is generated. A robot attempts to determine whether this obstacle is another robot or some passive object. To do this, it waits about 200ms the collision-avoidance signals of other robots in the environment. These received IR-signals indicate the locals presence of another robot (the robot itself also emits such signals during the detection phase). When there are no such signals, this indicates a passive object, which cannot emit IR signals. When the collision contact is another robot, the local light intensity will be measured. The robot reads several values from both sensors and then average them. Depending on the received value (through interruption), the robot sets its timer according to the curve depicted in Figure 17 to wait up to 50-60 sec. without motion, but still emitting IR signals to be identifiable as a robot. This allows the creation of a seed point for further clusters. After this time the robot turns back to the cluster/robot and starts a motion again. When there are other collision contacts, the whole procedure is repeated. In
this way, the robot remains blocked within the cluster.

We performed two series of experiments: the preliminary experiments, where we looked for appropriate sensors (see Figure 11a) and the final experiments, shown in Figure 19. The final experiments included series of tries with 1 to 18 robots, equipped with the extension sensors board, as shown in Figure 12b. As in the preliminary experiments, we used a round luminescent lamp, which was mounted about 40-50cm above the arena. The ambient light was also generated by tube-luminescent lamps about 3m above the arena. The size of arena was 140 \times 115cm. Figure 20 illustrates the images from an experiment using 3 robots. It is clear that there was no clustering at the target location. In follow-up experiments, we increased the number of robots step-by-step. The critical swarm density, where aggregation at the target place started, was reached with 9 robots. In Figure 21 we show pictures from an experiment with 15 robots in which the robots aggregated quickly at the target location. In further experiments we moved the lamp into different positions of the arena, the robots disaggregated and re-aggregated at the new location. A video of these final experiments can be downloaded from the www.swarmrobot.org.

Figure 19: Final experiments with 1-18 micro-robots.
Figure 20: A group of 3 robots navigating with our algorithm in the arena. They were not able to find the place with the optimal illumination, which was located in the upper left corner of the arena.

Figure 21: A group of 15 robots navigating with our algorithm in the arena. They quickly and collectively found the optimal spot, which was located in the upper left corner of the arena.
4.1 Aggregation time and scalability of robot behavior

During preliminary, final and post-final experiments we performed over 100 tries (the final experiment with 18 robots was used as a demonstrator in museum). Additionally, there were also performed a few post-final experiments with extra large robot arena (3 × 3 meters) and the number of robots 105, 75, 50, 35 and 25.

Figure 22: (a) Extra large robot arena (3 × 3 meters) used in post-final experiments; (b) Large-scale swarm (135 robots) in post-final experiments.

After performing these experiments we identified three main factors influencing the aggregation. The first one is the number of collision contacts $n_s$ defined by the relation between swarm density $D_s$, perception radius $R_s$ and velocity of motion $v$, and expressed by (3). Increasing $n_s$ (for example by increasing $D_s$, $R_s$ or $v$) leads to more faster aggregation. When $n_s$ remains constant, performance of the aggregation is expected to be also constant (taking into account two other factors). The second factor is the area $S_{spot}$ and the position of the light spot. When $^2 S_{spot} < 0.75NR_s^2\pi$, robots have not enough place for aggregation and this factor could be an environmental bottle neck. Position of the light spot on the arena defines how quickly the seed point can be created. For example, 18 robots require about 1 min. for aggregation with the light spot in the corner area. In contrast, 15 robots requires only 20-30 sec. with the light in the middle of the arena. The last factor, which influences the performance and scalability, is the waiting time in the cluster. Robots in the middle of the cluster are blocked by other robots and therefore remain in cluster. However robots in boundary area leave the cluster and this leads to disaggregation. When

\[^2\text{We assumed that the aggregation is successful when about 75\% of N robots are located in one cluster under lamp(s).}\]
waiting time is too low, small initial clusters will always disaggregate, so that robots
do not find an optimum. However, when waiting time is too large, optimal cluster
will also not appear because non-optimal clusters will not disaggregate. The maximal
waiting time is experimentally chosen as 1 minute.

To test the scalability of the algorithm, both the number of robots (18, 25, 35,
50, 75, 105), the area of light spot (4025cm$^2$, 11250cm$^2$, 22500cm$^2$) and robot arena
(140 × 115cm$^2$, 300 × 300cm$^2$) are step-wisely increased. For 15-18 robots in arena
140 × 115cm$^2$ and 4025cm$^2$ light spot the aggregation time was between 30 sec. and
1 min. in almost all experiments (depend on the position of the light spot). The
aggregation time for 25 and 35 robots in the same arena and light spot is decreased
to 20-30 sec. (however in this case > 25% of robots moved outside of the light spot).
The aggregation time for the arena size 300 × 300cm$^2$ with 35 robots and area of the
light spot 11250cm$^2$ was about 2 min. Doubling the area of light spot (22500cm$^2$)
lead to more faster building of the seed point(s), however the averaged aggregation
time remained within 2 min. Aggregation time for 50 robots with the same conditions
was about 1 min. Further increasing the number of robots (75, 105) in the same area
of robot arena and light spot demonstrated fastening of aggregation up to 20-30 sec.
(however here also > 25% of robots moved outside of the light spot).

5 Conclusion

We believe that the proposed algorithm is one of the simplest possible optimum-
finding algorithms for swarm robotics (concerning algorithmic complexity, computa-
tional efforts and sensory abilities). In addition, this algorithm shows interesting
scaling properties because the inter-robot-collisions enhances collective performance
in contrast to usual algorithms where such collisions are usually seen as negative
counter-productive events. As shown by final and post-final experiments, the robot
system with this algorithm belong the class of super-scalable systems (Constantinescu
et al., 2004). Interestingly, the algorithm shows that these robot-to-robot interactions
are essential for achieving the common goal, but the algorithm does not involve any
robot-to-robot communication which means no messages are passed from one robot
to another. So an almost paradox (at least a counter-intuitive) situation emerges: a swarm of non-communicating robots works more efficiently in bigger groups than in smaller ones. In addition, the algorithm has all the properties that characterize "swarm intelligence". These include: emergence, flexibility, robustness against perturbation, robustness against initial conditions, and robustness against sub-optimal solutions (local optima).

Another important result is the systematic re-embodiment of behavioral patterns from natural agents into artificial ones. Through re-embodiment we can keep the structure of the original approach and in this way we can keep its efficiency and scalability. However, after performing the preliminary and main experiments, it was evident to us, that even this simple aggregation algorithm requires specific sensor/actor systems: the behavioral mechanisms of collision-avoidance in dense clusters should be matched with the creation of seed points and in turn with the resting time. The resting time itself is a function of temperature. In this way, the actuation is closely related with the sensor input. The reproduction of natural sensors-actor couplings into artificial ones is the most important step of the re-embodiment procedure. We demonstrated that the "virtualization" of sensor/actor systems, where for example temperature was replaced by light and chemical recognition by passive sensing of the IR-light, can be successful in terms of their functionality and efficiency. However, this step is critical to the technological restrictions imposed on real robotic sensors and actuators. Since biological sensing and locomotion is much more "advanced" than the current state of the art in robotics, the successful re-embodiment of biological behavior in a robot represents a trade off between algorithms and hardware development.

Re-embodiment raises two further questions that still remain yet unanswered. The first point goes to the structure of behavior rules shown in Figure 10. We see that each behavioral pattern is developed and optimized for specific sensor-actor system (which also includes a sensor data preprocessing). We assume that the most complex parts of systems that generate behavioral patterns are dedicated to the sensor-actor coupling, at least this appears evident in the robotic case. We ask ourselves about the existence of a "hardware-free" behavioral pattern, which should represent a kind of "pure intelligence", being free from any limitations. The question is whether or not
we will be able in future to derive such a "pure intelligence" by means of sensor/actor "virtualization"?

The second question is about bio-inspired and tech-inspired research. The experiments were clearly motivated by the observation of bees. However, during the experiments with robots we identified several unexplained phenomena, where bees behave differently than the robots (e.g. cluster re-joining). This different behavior points to some latent factors, which further experiments with bees should clarify. In this case we see a feedback to biological experiments which can be denoted as "tech-inspired". Both bio- and tech-inspired research can supplement each other in better understanding collective intelligence emerged from natural and artificial systems.

References


