Collective Perception in a Robot Swarm

Thomas Schmickl[1], Christoph Möslinger[2], and Karl Crailsheim[1]

[1] Department for Zoology, University of Graz, 8010 Graz, Austria
schmickl@nextra.at
[2] FH St. Pölten, 3100 St. Pölten, Austria

Abstract. In swarm robotics, hundreds or thousands of robots have to reach a common goal autonomously. Usually, the robots are small and their abilities are very limited. The autonomy of the robots requires that the robots' behaviors are purely based on their local perceptions, which are usually rather limited. If the robot swarm is able to join multiple instances of individual perceptions to one big global picture (e.g. to collectively construct a sort of map), then the swarm can perform efficiently and such a swarm can target complex tasks. We here present two approaches to realize 'collective perception' in a robot swarm. Both require only limited abilities in communication and in calculation. We compare these strategies in different environments and evaluate the swarm's performance in simulations of fluctuating environmental conditions and with varying parameter settings.

1 Introduction

In robot swarms, hundreds or thousands of small and simple robots have to perform in a well-organized and efficient way to pursue common goals. With increasing size of the swarms, external controllers that have a 'global view' of the swarm's environment get inefficient because the control of each single robot within the swarm gets intractable even for strong computers. Also pre-calculated plans represent no solution with swarm sizes beyond a few hundred robots. Another problem is the inter-robot communication in such huge swarms, because if every robot has to communicate with every other robot, the required width of the communication channel increases non-linearly with the swarm size. In the I-SWARM project [1][2][3], we want to implement a swarm of very small robots (approx. 8mm³ size) that is able to perform collective perception. To us, the term "collective perception" describes a way that allows taking advantage at the global (swarm) level from a mass of complex data sensed in parallel on the individual level. The final swarm decision is made at the conceptual level by a group of collaborative agents. This ability can enhance the performance of a swarm (e.g. optimize patch selection for foraging tasks [4]) and expands its range of application. The I-SWARM robots have only limited sensorial abilities and can communicate only at short distance by LED's and photodiodes. These restrictions create a demand for simple solutions of collective perception strategies.

Animal swarms demonstrate that a set of relatively primitive individual behaviors enhanced with local communication can produce a large set of complex swarm behaviors. Such animal swarms show self-organization [5] and swarm-intelligence [6][7]:

E. Şahin et al. (Eds.): Swarm Robotics Ws, LNCS 4433, pp. 144–157, 2007.

Bacteria, ants and bees are able to choose the optimal feeding site and to recruit an appropriate fraction of foragers to each food site. Ants use pheromones to manage this decision making collectively. Honeybees use a variety of dances performed near the hive entrance to choose their feeding sites and to recruit the appropriate number of forager bees and food-storage bees. In both cases, individual animals do not visit several feeding sites and do not compare them individually. In contrast, pheromones and dances generate a structured environment that is regulated by positive and by negative feedback loops. These specialized environments act like 'maps' that are built up collectively and that are 'read' by many individuals in parallel. The most fascinating examples of 'collective perception' are found in honeybees. Forager bees and storage bees evaluate simple cues like queuing delays [8][9][10], searching times for empty combs [11] and multiple nectar transfers [12] to assess the current global workload balancing, the global need for comb construction and the environmental nectar flow.

Our approach to a bio-inspired technique for collective perception in swarm robotics is inspired by one of these examples of 'collective perception' in honeybees: By evaluating trophallactic contacts[1] forager bees can indirectly assess the current ratio of brood demand to pollen supply in the colony without inspecting brood area and pollen stores individually [13][14][15]. Nurse bees eat and digest pollen to derive a proteinaceous food (jelly) from it [16]. This jelly is fed to larvae and is exchanged frequently among adult bees. In times with high pollen demand, when a lot of brood has to be fed, the larvae consume the main part of the proteins, so that forager bees do not receive high amounts of proteins through trophallaxis. It is assumed that foragers are therefore more "protein hungry" and are more likely to forage for pollen instead of nectar. This way, the colony responds to a high pollen demand by recruiting more foragers to pollen collecting. The collective of forager bees indirectly perceives the current ratio of brood to food. In addition to proteins, the brood also consumes large amounts of nectar and nectar is also passed from bee to bee via trophallaxis.

Our goal was to use mechanisms in our robot swarm that are as simple as the biological examples mentioned above. We tested two approaches, one is a rather technical solution and was already used in swarm robotics, and the other approach is inspired by the trophallactic interactions of honeybees. Both methods are compared in the same simulated environment. The bio-inspired strategy is evaluated in detail and the importance of its parameters is analyzed in detail. Finally, the bio-inspired approach had to demonstrate its advantages in a fluctuating environment.

2 The Scenario

In the experiments described here, we used our simulation platform LaRoSim (<u>La</u>rge <u>Ro</u>botswarm <u>Sim</u>ulator), which we already described in [17][18]. The simulator is a multi-agent simulation of approx. 1000 robots that move in an arena. These robots can communicate by LED's and photodiodes and can also sense walls and obstacles this way. In addition to that, special (color) marks on the floor can be sensed, but only if the robot is located directly above such a mark.

[1] Trophallaxis is the mouth-to-mouth transfer of fluid food between adult honeybees.

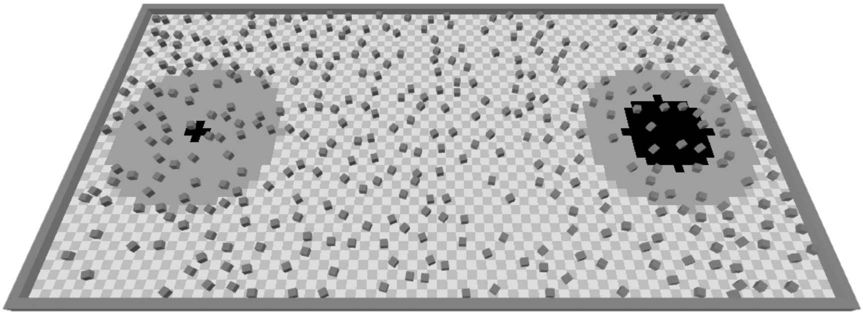


Fig. 1. A screen shot of our simulation platform LaRoSim. The two black areas (small left and huge right) represent target areas for aggregation. The gray circles indicate the zones in which we counted the robots for evaluating the aggregation success. Gray boxes represent robots.

Figure 1 shows a screen shot of the scenario the robot swarm has to perform in. Two black marks indicate aggregation areas (e.g., places to work). These areas can be of different sizes. The goal of the swarm is:

1. Explore the arena to detect these target sites.
2. Communicate the location of the targets to the other robots, so that they can aggregate there.
3. Recruit cohorts of robots to each target. The sizes of these cohorts should correspond to the size of the target areas.

In conclusion, the robot swarm has to manage to measure and to compare the sizes and the distances of the two target areas collectively. This goal can only be achieved collectively, because it goes far beyond the sensorial capabilities of a single robot. We chose a very simple example of work that has to be performed by the robot swarm (pure aggregation), because we wanted to concentrate on the problem of 'collective perception' in this study. More sophisticated work in LaRoSim, e.g. collective floor cleaning and optimal route finding, was already shown in [17][18]. To evaluate the

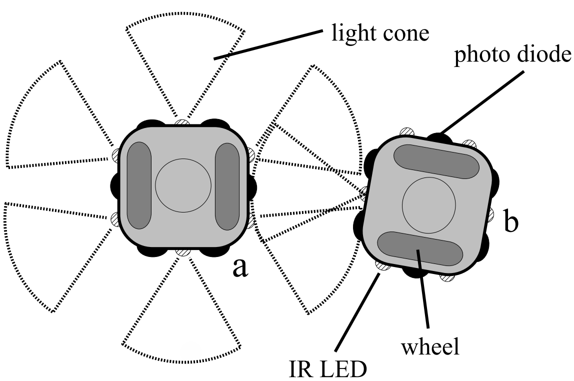
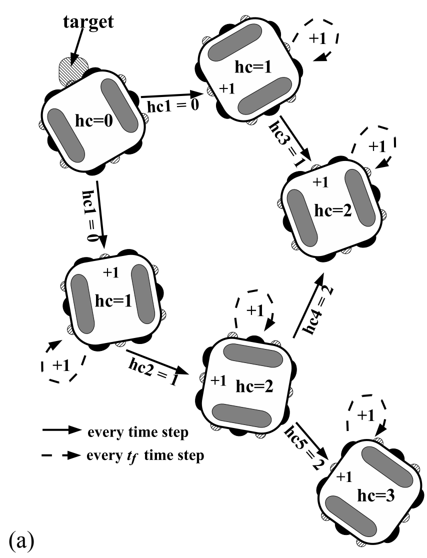


Fig. 2. Morphology of the robots in the simulation environment. In the picture, the two robots can establish a bi-directional communication, because one receptor of each robot is within the light cone of the other robot.

recruitment of robot cohorts to the two targets, we measured the number of aggregated robots in a radius of 10 robot-diameters (rd) around the center of each target area, as indicated by the gray circles around all black target areas in figure 1. Please note, that the robots have no ability for long-distance communication and no long-range sensing for target areas. The information about the location of the target areas has to be propagated through the swarm by using only nearest-neighbor communication, as depicted in figure 2. The communication radius is 3.5 rd.

2.1 The Hop-Count Strategy

The first strategy that we implemented in our robots is called 'hop-count' strategy. This strategy works as follows: The robots move randomly and try to avoid collisions and walls. Each robot **i** has an internal memory **hc(i,t)** that is set to the maximum possible hop-count **hc(i,t)=**$\mathbf{hc_{max}}$. If a robot encounters a target area, it sets **hc(i,t)=0**. During the run, all robots communicate with their nearest neighbors within their communication radius. The focal robot **i** compares its own hop-count with every neighbor **j**. If the neighbor has a lower hop-count (**hc(j,t)<hc(i,t)**), robot **i** copies the hop-count value of the neighbor and increases it by **1**. Every $\mathbf{t_f}$ time steps, the robot **i** increases its hop-count value by **1** spontaneously (**hc(i,t)=hc(i,t-1)+1**). This process is called 'forgetting', because it forces wrong or out-dated information to leave the system over time. If **hc(i,t)** exceeds $\mathbf{hc_{max}}$, **hc(i,t)** is set to $\mathbf{hc_{max}}$. This way a gradient emerges within the robot swarm that points downhill to the target areas. A robot that experiences a neighbor with



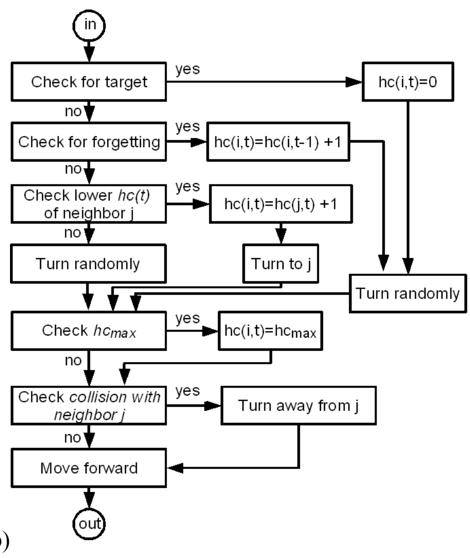


Fig. 3. (a) The gradient of hop-counts that emerges in the 'hop-count' strategy. The robot on the target sets its hop-count to 0. All robots copy the lowest hop-count from their neighbors and increase it by 1. After some ($\mathbf{t_f}$) time steps, they increase the hop-count spontaneously ('forgetting'). (b) Behavioral program of a robot in the 'hop-count' strategy. This program is executed every time step.

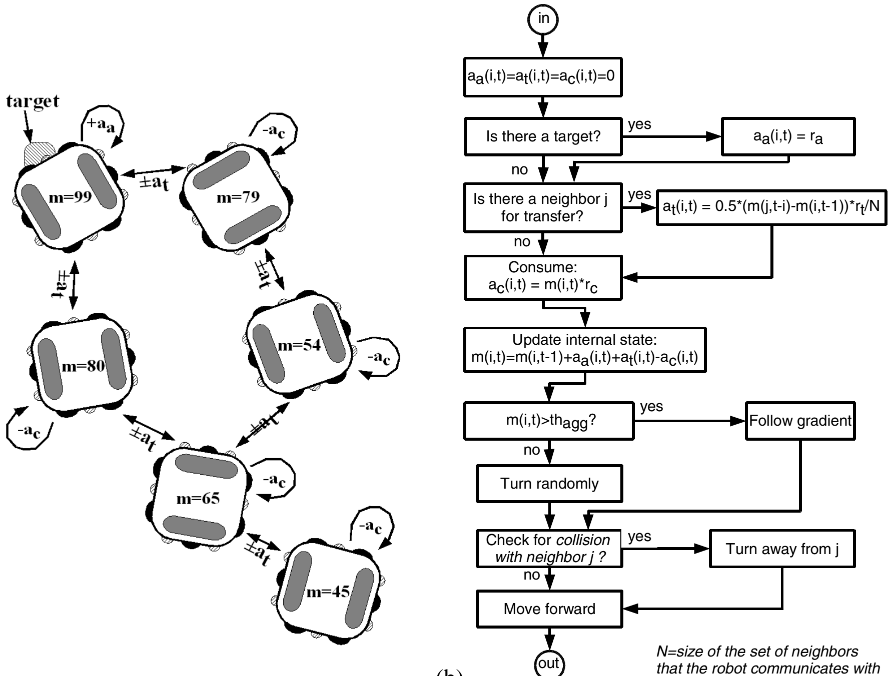
a hop-count that is smaller than or equal its own hop-count navigates towards this neighbor. If more than one neighbor has the same low hop-count, the robot calculates its direction by averaging the vectors towards these neighbors. Figure 3a depicts the emergence of the gradient within the robot swarm. Figure 3b shows the behavioral program that is executed by each robot at every time step.

2.2 The 'Trophallaxis-inspired' Strategy

The 'trophallaxis-inspired' strategy is inspired by a behavior that is frequently found in social insects: The mouth-to-mouth transfer of liquid food between adult animals. In honeybees, beekeepers often install feeders in the hives to provide the bees with sugar-water. At these feeders, some bees fill their crops and then move away. On their way through the hive, they meet other bees and can share parts of their nectar load with them. It is assumed, that the more nectar the donor bee has and the less nectar the receiver bee has, the more nectar is transferred on average. On their way, the bees also consume a fraction of their nectar load to gain energy from it.

In the robot-swarm, the nectar crop of the bee is represented by a memory place inside of the robot. Basically each robot $\mathbf{i}$ starts with random movement and with a memory value $\mathbf{m(i,t)=0}$. If the robot encounters a target, it adds a defined amount of 'virtual nectar' to its memory $\mathbf{a_a(i,t)=r_a}$ ($\mathbf{r_a}$: addition-rate, $\mathbf{a_a(i,t)}$: amount of addition). Every time step, robot $\mathbf{i}$ communicates with its local neighbors $\mathbf{j}$ and exchanges an amount of 'virtual nectar' with them. The amount $\mathbf{a_t(i,t)}$ of this exchange is proportional to the differences in the memory values among the robots and is determined by the transfer-rate r_t: $\mathbf{a_t(i,t)=0.5*(m(j,t\text{-}1)\text{-}m(i,t\text{-}1))*r_t/N}$. The variable $\mathbf{N}$ represents the number of local neighbors the focal robot communicates with. In case of $\mathbf{N=0}$, the value of $\mathbf{a_t(i,t)}$ is set to $\mathbf{0}$. Every time-step, each robot $\mathbf{i}$ also decreases its memory value by an amount $\mathbf{a_c(i,t)}$ which is defined by the consumption rate $\mathbf{r_c}$. $a_c\mathbf{(i,t)= m(i,t\text{-}1)*r_c}$. After all these in-flows and out-flows of 'virtual nectar' are calculated by each robot the memory-value can be updated according to the following equation: $\mathbf{m(i,t)=m(i,t\text{-}1)+a_a(i,t)+a_t(i,t)\text{-}a_c(i,t)}$. Please note that the 'trophallaxis-inspired' strategy uses floating point numbers, while the 'hop-count' strategy uses integer values only. By the rules mentioned above, again a gradient of memory values emerges within the robot swarm. If a robot $\mathbf{i}$ reaches a memory value above a threshold ($\mathbf{m(i,t)>th_{agg}}$), the robot turns towards its local neighbor with the highest memory value. If the memory value is below or equal $\mathbf{th_{agg}}$, the robot $\mathbf{i}$ moves randomly.

Figure 4a depicts how the gradient of 'virtual nectar' emerges in the robot swarm in the 'trophallaxis-inspired' strategy. Figure 4b depicts the behavioral program that is executed by every robot in every time step. In order to adjust the aggregation-sensitivity of the swarm we implemented a behavioral threshold $\mathbf{th_{agg}}$. A robot will only follow the gradient if its memory value is above the threshold $\mathbf{m(i,t)>th_{agg}}$. If its memory value is below or equal $\mathbf{th_{agg}}$, the robot will move randomly. Figure 4a depicts how the gradient of 'virtual nectar' emerges in the robot swarm in the 'trophallaxis-inspired' strategy. Figure 4b depicts the behavioral program that is executed by every robot in every time step.



(a) (b)

Fig. 4. (a) The gradient of 'virtual nectar' that emerges in the 'trophallaxis-inspired' strategy. The robot at the target adds 'virtual nectar' to its memory. All robots exchange fractions of the 'virtual nectar' proportionally to the inter-robot differences. All robots consume 'virtual nectar' over time, thus they decrease their memory values ('forgetting'). (b) Behavioral program of a robot in the 'trophallaxis-inspired' strategy. This program is executed every time step.

3 Results

In our simulation runs, both strategies were able to produce the desired aggregation behavior at the target areas. But this was not the main focus of this study. The main question was, whether or not the swarm will be able to collectively measure the sizes of the target areas and to proportionally recruit the appropriate number of robots to these targets.

3.1 Scaling the Sizes of the Target Areas

In this experiment, we tested both strategies in environments with varying differences in the size of the target areas. The sizes of the targets areas were defined by their radii. We tested the following ratios of radii: 1:5, 2:4, 3:3, 4:2, and 5:1. We started 375 robots that were (uniformly) randomly distributed within the arena. The results of these simulation runs are depicted in figure 5. The 'hop-count' strategy recruited more robots during the runtime of the experiments (=250 time steps) than the other strategy, but failed to recruit the robots according to the target sizes. The aggregation was measured by counting the number of robots within a radius of 10 robot-diameters around the center of each target (see figure 1). For the simulation runs, we used the following parameters: $\mathbf{r_a=50}$, $\mathbf{r_c=0.01}$, $\mathbf{r_t=1}$, $\mathbf{hc_{max}=40}$, $\mathbf{t_f=5}$. The aggregation threshold $\mathbf{th_{agg}}$ was set to 100. For collision avoidance, the robots tried to stay away from each other half of their communication radius (**coll-dist=0.5**). Robot speed was 0.25 robot-diameters per step. The trophallaxis-inspired strategy recruited lower robot numbers but managed to recruit the robots accordingly to the

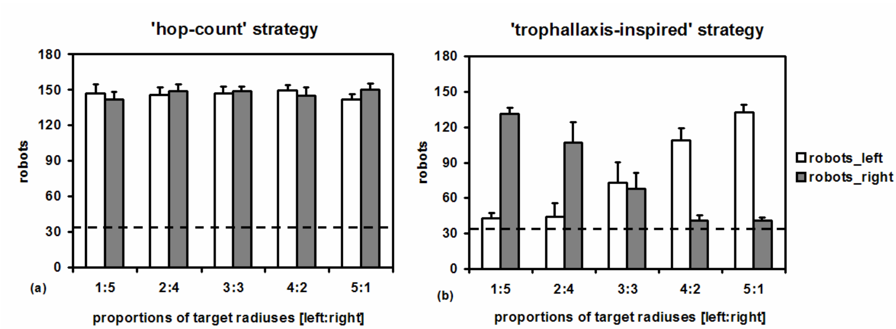


Fig. 5. Collective decisions made by the robot swarm in different environments. The dashed line shows the expected number of robots that would have been in the measurement area (radius=10 each) if there had been no aggregation behavior at all. N=10 per setting. Bars represent mean values and whiskers indicate standard deviations. Duration: 250 time steps.

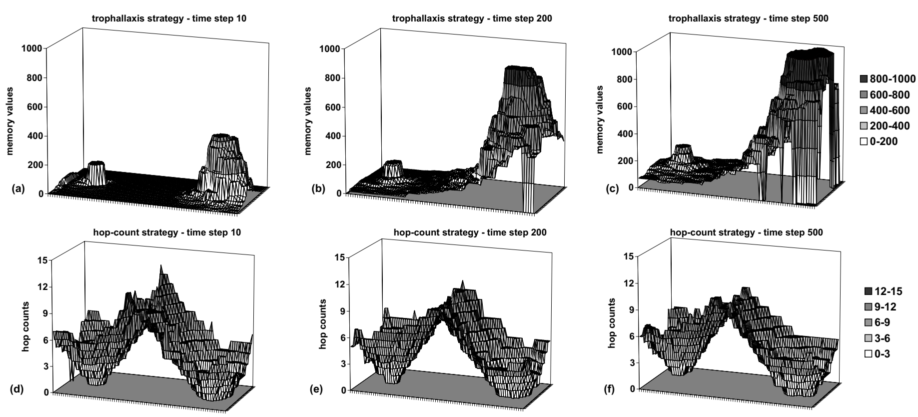


Fig. 6. The dynamics of the emerging gradients in our experiment. (a-c): The dynamics of the gradient in the trophallaxis inspired strategy. For generating the picture, we assigned the maximum memory value of all visible robots to each location in the arena. (d-f): The dynamics of the gradient in the hop-count strategy. Here we assigned the minimum hop-count of all visible robots to each position in the arena. Both simulation runs used extreme environmental conditions: The left target was very small (radius=1) and the right target was large (radius=5).

sizes of the target areas. An explanation for these results can be found in figure 6, which depicts two simulation runs with very extreme conditions: A small target on the left side (radius=1) and a huge target on the right side of the arena (radius=5). The hop-count strategy generates two bowl-shaped gradients that immediately reach the whole arena. The two bowls are of almost equal size and so the recruited cohorts of robots were also of almost equal size. In the trophallaxis-inspired strategy, the emergence of the gradient is much slower. But the bigger target on the right side allows more robots to add 'virtual nectar' to the system through their addition- and transfer-rates. This leads to a much higher 'mountain' that is able to recruit the majority of the robots to the right side. Obviously, the hop-count strategy is only able to report the distance of the target to other robots, while the trophallaxis-inspired strategy is able to report also the sizes of the targets.

3.2 The Importance of the Swarm Density

In swarm robotics, the swarm density is an important factor. To test how swarm densities affect the abilities of swarms to perform collective perception we further investigated the experiment with the biggest difference in target sizes (radii left:right = 1:5).

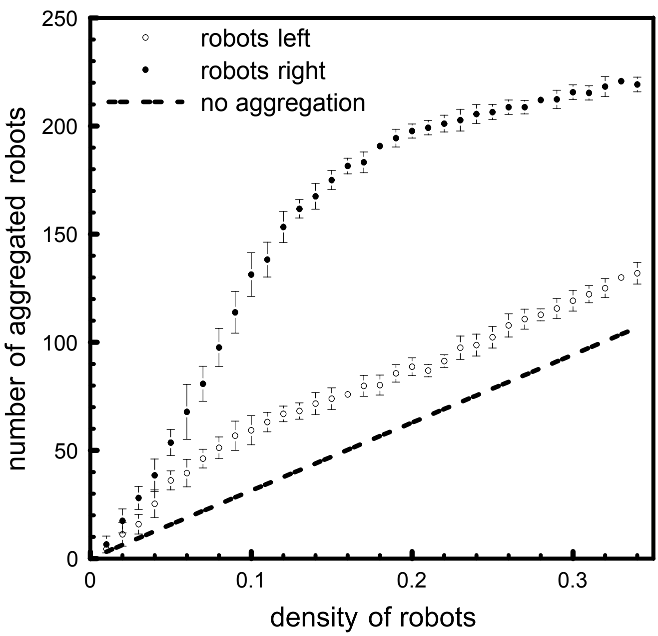


Fig. 7. Aggregation of robots to the small left target area (radius=1) and to the large right target area (radius=5) with varying swarm densities. The dashed line shows the expected number of robots that would have been in the measurement area (radius=10 each) if there had been no aggregation behavior at all. N=10 per setting. Duration: 250 time steps.

We only tested swarms using the trophallaxis-inspired strategy because swarms using the hop-count strategy couldn't differentiate between target sizes (see sub-section 3.1, figures 5,6). For the following analysis we used the same parameter settings as we used in sub-section 3.1. The only varied parameter was the density of the robots, which we scaled between **0.01** and **0.34**, which corresponds to swarm sizes of **30** robots and **1047** robots. Figure 7 shows the results of these experiments: Aggregation was performed on both target areas. With a swarm density of **0.17**, the maximum preferential aggregation was found at the large target. With higher densities (**> 0.2**), no increase in aggregation is found anymore, the number of robots increases linearly as a product of pure random walk (dashed line). This analysis was made with a value of $\mathbf{th_{agg}=-50}$ to demonstrate that with the trophallaxis-strategy the swarm can also perceive small target areas (see section 3.3 for details). With $\mathbf{th_{agg}=0}$, no aggregation on the small target size can be observed (data not shown), the number of robots around the small target is predictable by considering solely the random walk.

3.3 The Role of the Aggregation Threshold (th_{agg})

The results of the experiments in subsection 3.1 demonstrate that in the trophallaxis-inspired strategy, the huge gradient that emerges from the large target area increases over time and dominates over the gradient emerging at the location of the small target area. Nevertheless, the small target also recruited a few robots (see figure 5). By adjusting the threshold $\mathbf{th_{agg}}$ we were able to indirectly determine the minimum target size that lead to aggregation. In our strategy, the strength of aggregation was regulated by the variable **weight(i,t)**, which represents the ratio of directed movements to random movements. Robots with a low memory value **m(i,t)** have a low **weight(i,t)** and thus they perform a random walk most of the time, whereas robots with a high memory value **m(i,t)** have a high **weight(i,t)** and will move towards the target in a very directed way. Thus threshold $\mathbf{th_{agg}}$ is used as an offset in our computation of **weight(i,t)**. For example, with negative values of $\mathbf{th_{agg}}$ we can achieve a more directed movement of robots with a low memory value **m(i,t)**. Figure 8 depicts the dependency of the variable **weight(i,t)** on the variable **m(i,t)** and on the parameter $\mathbf{th_{agg}}$.

$$weight(i,t) = \max\begin{cases} \min\begin{cases} \dfrac{m(i,t) - th_{agg}}{1000} \\ 0.75 \end{cases} \\ 0 \end{cases} \tag{1}$$

In the following experiment, we wanted to test, whether or not an adjustment of the threshold $\mathbf{th_{agg}}$ can modulate the sensitivity of the swarm for smaller target areas. Figure 9 shows the results of this experiment: Between $\mathbf{0 < th_{agg} < 300}$, the aggregation at the large target is negatively correlated with the value of $\mathbf{th_{agg}}$, the small target is almost ignored by the swarm. With negative values of $\mathbf{th_{agg}}$, the aggregation at the small target increases significantly, without affecting the aggregation at the large target area. This shows that adjustment of $\mathbf{th_{agg}}$ leads to recruitment of previously non-recruited robots around the small target.

3.4 The Role of the Negative Feedback ($\mathbf{r_c}$)

In swarm robotics, the decay of information is important as soon as the swarm of robots has to act in changing environments. It is needed to allow out-dated, thus not reinforced, information to leave the system. In the trophallaxis-inspired strategy, this is achieved by a constant consumption of 'virtual nectar'. If a target area disappears, there will be no local addition of 'virtual nectar' and the gradient will disappear. To investigate this, we performed an experiment with very extreme differences in target sizes (radii left:right = 1:5). After 500 time steps, we changed the sizes of the targets:

The big area got small and the small area got big (radii left:right 5:1). After the same time span, we investigated how the swarm responded to this fluctuation by counting the newly recruited robots at the left target and the robots that abandoned the right target after 1000 time-steps.

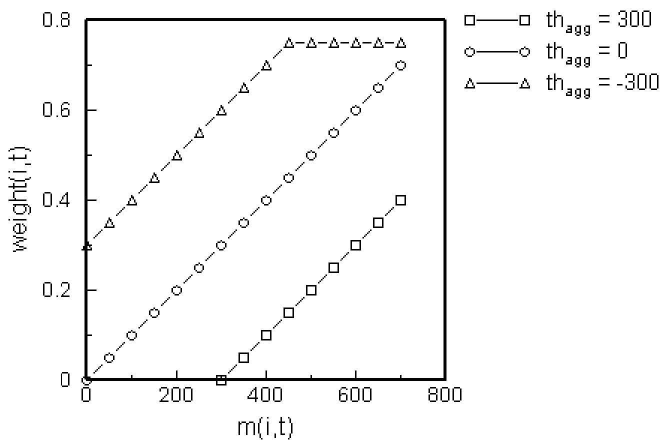


Fig. 8. Values of **weight(i,t)** as a measurement for directedness of a robot's movement depending on its memory value **m(i,t)** and the threshold th_{agg}. Shown for a positive threshold $\mathbf{th_{agg}}$=300, no threshold $\mathbf{th_{agg}}$=0, and a negative threshold $\mathbf{th_{agg}}$=-300.

We initially implemented the 'forgetting' also into the hop-count strategy ($\mathbf{t_f}$), but this strategy failed to recruit proportional cohort to differently sized target areas in a stable environment (sub-section 3.1, figures 5,6). Without such a proportional response of the swarm, it is useless to perform such a test in a fluctuating environment, so we only analyzed the trophallaxis-inspired strategy here. We kept all parameter settings identical to the runs shown in subsection 3.1, but we varied the values of the

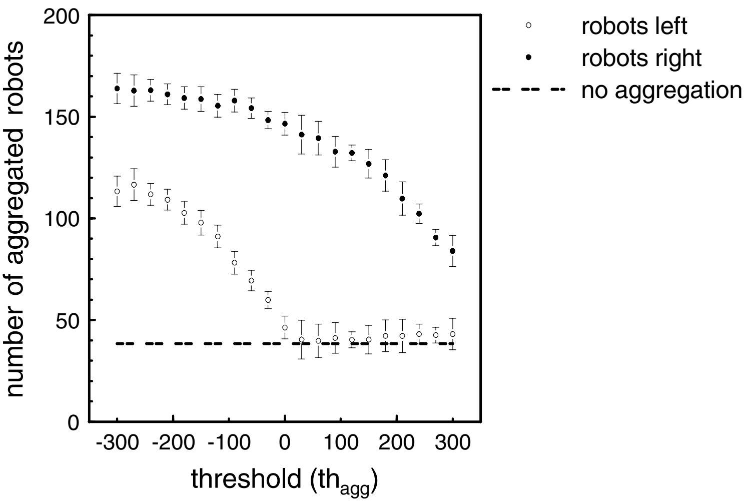


Fig. 9. Aggregation of robots to the small left target area (radius=1) and to the large right target area (radius=5) with varying threshold (**th$_{agg}$**). The dashed line shows the expected number of robots that would have been in the measurement area (radius=10 each) if there had been no aggregation behavior at all. N=10 per setting. Duration: 250 time steps.

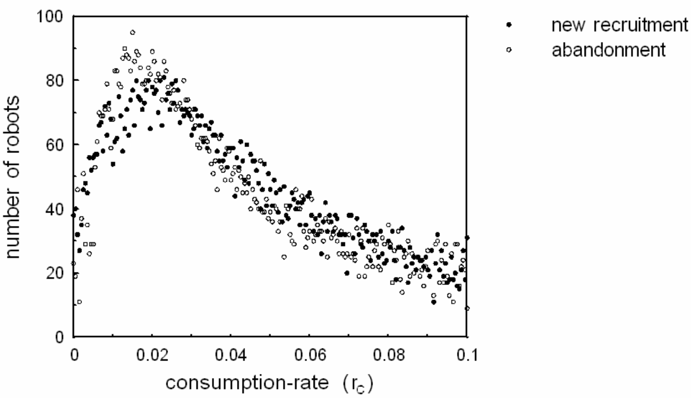


Fig. 10. New recruitment and abandonment of robots in a changing environment and with varying values of the consumption-rate (**r$_c$**). High values of new recruitment and of abandonment indicate a high flexibility of the collective decisions of the robots swarm. N=1 per setting. Measurements were made 500 time steps after the environmental fluctuation.

consumption rate **r$_c$** between 0 and 0.1. So we compared never-forgetting swarms, moderately fast forgetting swarms and quickly forgetting swarms. Figure 10 shows the results of this experiment: Never-forgetting swarms (**r$_c$**< 0.01) failed to adjust to the switch because the strong gradient that had emerged around the right target before the switch kept dominating throughout the arena. Quickly forgetting swarms (**r$_c$**>0.03) on the other hand were not able to establish a gradient that reached robots that were far from the target and thus changes in the environment were not noticed by most robots. With a consumption-rate between 0.01 and 0.03, the swarm showed the highest flexibility in its decisions.

4 Discussion

Our simulation experiments focused purely on the questions of collective perception in a robot swarm. We showed that a system that exploits purely 'hop-counts' of messages is able to navigate robots to target areas but fails to perform a collective perception of target area sizes. Such 'hop-count'-based strategies were used (and published) several times in swarm robotics [19][20][21][22]. Some times these techniques are called 'virtual pheromones', a term that we (as biologists) do not think is appropriate. A pheromone is a chemical substance that is released by an animal in the environment and that causes a behavioral change or a physiological change in a receiving animal. For a swarm robot, it is very difficult to deposit something in the environment; therefore hop-counts that are communicated from robot to robot are often used to mimic pheromone gradients. But such a system has significant differences to real pheromones, because hop-count values do not remain in place in the environment, they move with the robot that carries it. We think that these hop-counts and also the memory-values used in our trophallaxis-inspired strategies have much more analogies to the crop loads of (social) animals. They are bound to their 'carrier'-animals and it is often found in nature, that crop volumes are transferred from one animal to another (ants, termites, bees, wasps, birds, vampire bats). In contrast to the hop-count strategy, the trophallaxis-inspired strategy [17] was able to perform collective perception successfully (figure 5,6). By using this method, the swarm was able to collectively measure the size of the target areas and to communicate these sizes throughout the swarm.

Please note that a single robot cannot measure the size of the target area, it can only determine whether or not it is located on a target area. The observed effect is caused by the fact that a larger target area can contain more robots and thus more 'addition' is made to the system. The three parameters 'addition-rate', 'transfer-rate', 'consumption-rate' can be used to regulate the system. A higher addition makes the gradients higher. The transfer rate allows the gradient to reach further, thus it can be used to regulate the range of the attraction of the targets. We showed that in changing environments, a moderate forgetting of collective perceptions plays an important role. With a consumption-rate that was too low, the robot swarm was not able to re-decide after the environmental fluctuation. With a consumption-rate that was too high, the swarm was not able to perform any collective decision at all. The threshold $\mathbf{th_{agg}}$ is an important factor to adjust the 'collective sensitivity' of the robot swarm. By adjusting this parameter, smaller target areas can be made invisible for the swarm, so that it focuses on the bigger target areas first. Our scenario (and the strategy) can be extended in several ways. In honeybees, the brood acts as a sink for food. In the case shown here, we used only target areas that led to an addition of 'virtual nectar'. If the scenario contains also areas that should be preferentially avoided (e.g., holes [23]), we could easily add such a sink to our system. Robots that encounter such areas reduce their memory values to 0. The threshold $\mathbf{th_{agg}}$ is currently a global parameter in our strategy. It will be interesting to introduce habituation and reinforcement to adjust this parameter individually, based on the prior work experience of a robot.

In conclusion, we demonstrated that collective perception of a robot swarm can be performed with simple nearest-neighbor communication, with rather narrow communication channels and with messages that include only little semantics. The system was shown to be robust, because our results were not significantly affected by random error (which we introduced in our simulation on motion, sensing and communication) or by initial conditions (robots were spread randomly in the arena). In addition, the collective decisions were flexible (see figure 10). Computational effort was low and the number of robots was rather high. All these features mentioned above indicate that the found collective perception was an emergent phenomenon of self-organization [5] and of swarm-intelligence [6][7].

Acknowledgement

This work is partially supported by: EU IST-FET-project ‘I-Swarm’, no. 507006.

References

1. Seyfried, J., Szymanski, M., Bender, N., Estana, R., Thiel, M., Wörn, H.: The I-SWARM Project: Intelligent Small World Autonomous Robots for Micro-Manipulation. In: Sahin, E., Spears, W.M. (eds.) Swarm Robotics. Springer LNCS 3342, (2005) 70 – 83
2. Kornienko, S., Kornienko, O. Constantinescu, C., Pradier, M., Levi, P.: Cognitive micro-agents: individual and collective perception in a microrobotic swarm. In: Proceedings of the IJCAI-05 Workshop on Agents in Real-Time and Dynamic Environment, Edinburgh, Scotland, (2005) 33 – 42
3. Kornienko, S., Kornienko, O., Levi, P.: Minimalistic approach towards communication and perception in microrobotic swarms. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, Alberta, Canada (2005) 4005 – 4011
4. Liu, Y., Passino, K.M.: Biomimicry of Social Foraging Behavior for Distributed Optimization: Models, Principles, and Emergent Behaviors. Journal of Optimization Theory and Applications, V 115, N 3, (2002) 603 – 628
5. Camazine, S., Deneubourg, J.L., Franks, N., Sneyd, J., Theraulaz, G., Bonabeau, E.: Self-organization in biological systems. Princeton University Press, NJ, USA (2001)
6. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm intelligence: From natural to artificial systems. Oxford University Press, New York, NY, USA (1999)
7. Kennedy, J., Eberhart, R.C.: Swarm Intelligence. Morgan Kaufmann Publishers, Academic Press, USA. (2001)
8. Anderson, C, Ratnieks, F.L.W.: Task partitioning in insect societies. I. Effect of colony size on queueing delay and colony ergonomic efficiency, Am. Naturalist 154, (1999) 521 – 535
9. Ratnieks, F.L.W., Anderson, C.: Task partitioning in insect societies. II. Use of queueing delay information in recruitment, Am. Naturalist 154, (1999) 536 – 548
10. Seeley, T., Towey, C.: Why search time to find a food-storer bee accurately indicates the relative rates of nectar collecting and nectar processing in honey bee colonies. Animal Behaviour, 47, (1994), 311 – 316

11. Pratt, S.C.: Optimal timing of comb construction by honeybee (*Apis mellifera*) colonies: a dynamic programming model and experimental tests. Behavioral Ecology and Sociobiology 46, (1999) 30 – 42
12. Huang, M., Seeley, T.: Multiple unloadings by nectar foragers in honey bees: a matter of information improvement or crop fullness? Insectes Sociaux 50, (2003), 1 – 10
13. Camazine, S.: The regulation of pollen foraging by honey bees: How foragers assess the colony's need for pollen, Behavioral Ecology and Sociobiology 32, (1993) 265 – 273
14. Camazine, S., Crailsheim, K., Hrassnigg, N., Robinson, G.E., Leonhard, B., Kropiunigg, H.: Protein trophallaxis and the regulation of pollen foraging by honey bees (*Apis mellifera* L.), Apidologie 29, (1998) 113 – 126
15. Schmickl, T., Crailsheim, K.: Inner nest homeostasis in a changing environment with special emphasis on honeybee brood nursing and pollen supply. Apidologie 35, (2004) 249-263
16. Crailsheim, K.: The flow of jelly within a honeybee colony, Journal of Comparative Physiology B 162, (1992) 681 – 689
17. Schmickl, T., Crailsheim, K.: Trophallaxis among swarm-robots: A biological inspired strategy for swarm robotics. In: Proceedings of BioRob 2006, Biomedical Robotics and Biomechatronics, Pisa, Italy. (2006) ISBN 1-4244-0040-6
18. Valdastri, P., Corradi, P., Menciassi, A., Schmickl, T., Crailsheim, K., Seyfried, J., Dario, P.: Micromanipulation, communication and swarm intelligence issues in a microrobotic platform. Robotics and Automation Systems (in press).
19. Payton, D., Daily, M., Estowski, R., Howard, M., Lee, C.: Pheromone Robotics. Autonomous Robots 11 (2001) 319 – 324
20. Payton, D., Estkowski, R., Howard, M.: Compound behaviors in pheromone robotics. Robotics and Autonomous Systems 44 (2003) 229 – 240
21. Stoy, K., How do construct dense objects with self-reconfigurable robots. In: Christensen, H.I. (eds.) European Robotics Symposium 2006, STAR 22 (2006) 27 – 37
22. McLurkin, J.D.: Stupid robot tricks: a behavior-based distributed algorithm library for programming swarms of robots. Master thesis at the MIT (2004)
23. Trianni, V., Nolfi, S., Dorigo, M.: Hole Avoidance: Experiments in Coordinated Motion on Rough Terrain. In: Groen, F, Amato, N., Bonarini, A., Yoshida, E., Krose, B., (eds.), Intelligent Autonomous Systems 8, (2004) 29-36