Optical Herding of Swarms: Toward Universal Control Algorithms for Microscopic Collectives

David Fielding[1], Imogen Taylor[1], Simon Jones[1], Sabine Hauert[1] and Edmund R. Hunt[1]

[1]Department of Engineering Mathematics, University of Bristol, Ada Lovelace Building, BS8 1TW, UK.
edmund.hunt@bristol.ac.uk

Abstract

Directed light beams are a promising means of control for microscopic agents, whether they are microrobots or phototatic microorganisms such as Volvox and ciliates. Given the simple reactive behaviors common to most microagents, there is likely to be a certain universality in light-beam algorithms that can usefully 'herd' such collectives around. Here, we develop three light-beam control algorithms to herd light-sensitive agents around a two-dimensional environment, each making varying assumptions about agent behavioral capacities. We test them with small swarms of Kilobot robots, which are about 3cm in size. These robots are convenient macro-scale demonstrators of possibilities at the micro-scale. The algorithms are tested in simulation and found to achieve the desired herding goals. Waypoint following missions were implemented using single robots and multiple robots to demonstrate more complex trajectories and highlight the effects of multiple robots interacting. One of the algorithms was tested with real robots and is shown to perform well, owing to good robustness to projection inaccuracies. Future swarm engineers could refer to a common toolbox of broadly effective light-based swarm control algorithms, which can be selected according to agent capabilities.

Introduction

The engineering of collectives or 'swarms' of microscopic agents will open up exciting new possibilities ranging from novel biomedical therapeutics (Alapan et al. (2019); Hauert and Bhatia (2014)) to functional materials with properties like self-healing (Slavkov et al. (2018)). Self-organization is usually key to adaptive collective behaviors in biological systems, whether engineered (Gorochowski et al. (2020)) or natural, and relies on reactive behavioral 'rules' that respond to local environmental conditions (Camazine et al. (2003)). Therefore, if a user could rapidly modify a self-organizing agent's environment, this could provide a method of control that would bypass the need for direct communication. Given the inherent challenges with engineering complex agents at the micro-scale, an environment-first approach could empower otherwise myopic reactive agents with purposeful, goal-directed movement.

Light is ideally suited for small scale environmental adaptation. It can power micromotors (Palagi et al. (2019)), alter shapes (Stoychev et al. (2019)), make or break chemical bonds (Chen et al. (2018)), force the release of a cargo (Erkoc et al. (2019)) alter micro-environments (Ruskowitz and DeForest (2018)) and influence light-sensing organisms (Jékely et al. (2008); Purcell and Crosson (2008)). Light has the capacity for high spatio-temporal resolution, and thus the simultaneous control of many agents, which contrasts with methods based on the use of chemicals or magnetic fields. Light-controlled microswarms can perform collective phototaxis (Dai et al. (2016)), self-assemble into active materials (Schmidt et al. (2019)) or even treat tumors (Tao et al. (2020)). Closed-loop high-resolution spatio-temporal control has been demonstrated in some cases, achieving collective behaviors such as flocking (Lavergne et al. (2019)), formation of complex patterns (Frangipane et al. (2018)) and group cargo transport (Steager et al. (2015)).

Herding or 'shepherding' of multiple coordinating agents is a recognized approach to solving the problem of simultaneously controlling many robots (Long et al. (2020)). Inspired by the use of sheepdogs to herd large numbers of sheep, one or more robotic 'shepherds' can be controlled to indirectly shape the movements of a group, as they move away from the shepherds toward a target location (Lien et al. (2004); Vo et al. (2009)). However, rather than control shepherd robot(s), here we manipulate the agents' environment directly. Depending on individual microagent capabilities, different light-based control algorithms could be relevant. An approach relying on agents avoiding light with random walk-type behavior makes quite minimal assumptions – this is our starting point for algorithm development.

Methods

Three algorithms were implemented and compared in simulation as well as with physical robots for one algorithm. Simulation provides a testing ground to develop proof of concept ideas and the ability to run repeated trials very rapidly. Practical experiments serve as a preliminary validation of one of the simulated algorithms.[1]

[1]Simulation, robot and analysis code can be found at: https://www.dropbox.com/s/wbkw8zao4sqljl9/CodeDrop.zip?dl=0

Simulation

The simulated arena consisted of N robots controlled by m magenta-colored projected regions aiming to reach an area of interest ('AOI') represented by a cyan circle. This could represent a target tumor site for microparticle treatment delivery, for example. The colors magenta and cyan were selected as these are the colors the physical robots can distinguish most consistently (sensing method described in Jones et al. (2018)). The simulation environment 'Kilobox' was developed by Jones (2019) and bundled with minimal example code which was modified and extended to achieve the desired functionality. The simulation speed can be modified and the GUI can be disabled via command line interface to reduce computational requirements when carrying out multiple runs. Bash scripts were written to automate repeated tests with different random seeds for data collection.

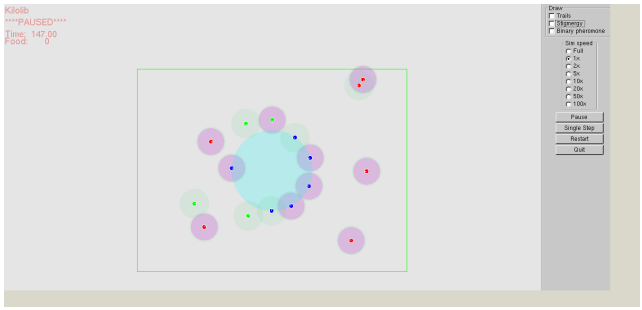


Figure 1: Kilobox Simulated Arena

Figure 1 shows the Kilobox simulation environment with multiple robots. The robots can change LED colors to indicate state changes: for all algorithms, robots will stop and turn dark blue when within the AOI. Magenta circles can be seen projected on the robots to control them.

Simulator Noise It is important to note for the following results that Kilobox models the intrinsic directional biases commonly found in real-world Kilobots by randomly initializing fixed biases of varying magnitude in linear and angular velocity. Real robots are rarely 'perfectly' calibrated, and even then suffer from inevitable asymmetries in e.g. motor behavior. The robot kinematics are described by

$$v = (m_L + m_R)/2 + v_\eta + v_k$$

$$v_\omega = (m_L - m_R)/L + v_{\omega\eta} + v_{\omega k}$$

where v is the linear velocity and v_ω is the angular velocity. m_L and m_R are the individual motor velocities; L the wheel (legs) separation; v_η and $v_{\omega\eta}$ are Gaussian noise terms injected when the motors are active, with $\sigma_{v\eta} = 0.001$ and $\sigma_{v\omega\eta} = 0.01$. v_k and $v_{\omega k}$ are biases fixed for the duration chosen at initialization from a Gaussian with $\sigma_{vk} = 0.0013$ and $\sigma_{v\omega k} = 0.0675$. These default values were chosen by measuring the behavior of multiple Kilobots (Jones (2019)). Outlier behavior in simulation can be seen as representing outcomes with poorly calibrated or imperfect robots.

Algorithms

Each method consists of two parts: first, the on-board algorithm controlling the individual Kilobot which determines how it reacts to locally sensed lighting; and second, the projected light region control algorithm, which shifts projected light regions according to different strategies. Here, we use robot-specific circles as control regions, but in future work we could use a wide variety of other light patterns: e.g., large rings, or 'highways', which may work well in some collective control cases.

The Kilobot algorithms consisted of a repeated loop every 0.5 s with a region color detection function and then a motor update; and in simulated robots a logging of spatial coordinates and time. The light control algorithms had a global view of all robots' positions, and bearings calculated from between timestep movements.

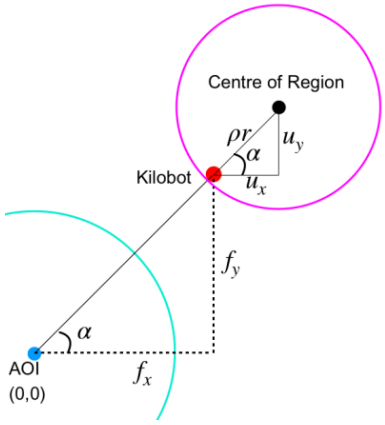


Figure 2: Algorithm 1 – Random Walk Avoidance

Algorithm 1: Random Walk Avoidance This uses random walk avoidance of circular light patches placed on the edge of the robot. It provides a performance baseline for comparison with two alternative algorithms. The region is placed so that the robot is positioned directly between the AOI and the region and just inside the region radius as shown in Figure 2. Robots move until no longer sensing the projected region, when they become idle. In our experiments each robot has its own separate region projected onto it one at a time, though multiple regions could also be projected concurrently.

Algorithm 2: Dynamic Biased Random Walk Avoidance This exploits random walk avoidance behavior again, but with an incrementally changing selective bias when light is applied. A circular light region is projected on the robot when its current bearing is wrong, and so the robot will decrease its probability of moving in its current forward direction. Conversely, the robot's probability of moving again in

the same direction is increased when it detects no control light. The robot will probabilistically select a direction (left, right or 'straight on') with a higher (lower) probability of going straight as the forward bias level increases (decreases) incrementally. The bearing error threshold for an intervention was set at $\pm\pi/8$.

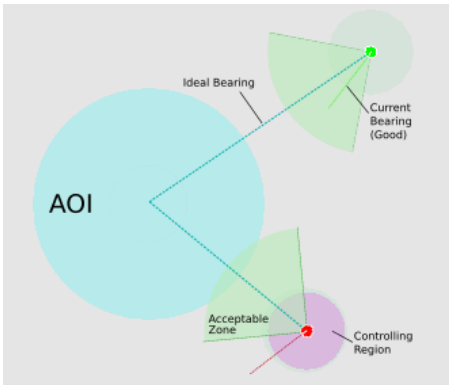


Figure 3: Algorithm 2 – Dynamic Biased Random Walk

Figure 3 illustrates the dynamic biased random walk method using two robots: one with a bearing within the acceptable threshold depicted in light green; and one greater than the threshold depicted in red being corrected by the controlling region.

Algorithm 3: Left-Right Alternation The third algorithm is a two-state controller which operates by measuring the difference between the current robot bearing and the ideal bearing. If the difference is negative then a circular light region is projected onto the center of the robot, triggering a rightward turn (clockwise). If the the difference is positive then no region is projected and the robot turns to its left (counterclockwise). In this fashion the robot zig-zags towards the AOI. Contrasting with the previous algorithms, there is no deliberate stochastic element.

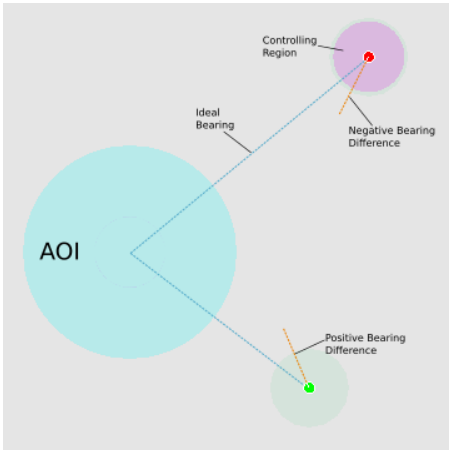


Figure 4: Algorithm 3 – Left-Right Alternation

Figure 4 illustrates the workings of the 'Light-Right Alternation algorithm' with two Kilobots. The red robot has a negative bearing difference and therefore the controlling region is projected to correct the bearing right while the green robot has a positive bearing difference and is 'instructed' to turn left by the lack of applied light region.

Practical Experimentation

Kilobots The robots used during the practical experimentation were Kilobots, small robots designed to be cheaply manufactured in order to make large robotic swarms attainable (Rubenstein et al. (2012, 2014)). A critical hardware component for this project was the ambient light detector to detect the projected light. Some demonstrations used multiple robots simultaneously; the completion time and path efficiency statistics were sampled from one well-calibrated robot used in multiple trials.

Experimental Setup The Kilobot robot positions were tracked using a HD Pro Webcam C920 with a resolution of 2304x1536 pixels placed above the robots. The regions were projected using a standard commercial BenQ MW632ST projector. The arena consisted of a 270 cm by 175 cm platform raised from the ground by 35 cm, supporting an enclosed smooth white acrylic sheet. The smooth surface was necessary for the robots' slip-stick locomotion. The projector and webcam setup allows for a closed control loop to direct the robots to the AOI. The arena setup is illustrated in Figure 5. This setup has been used to explore behavior tree evolution (Jones et al. (2018)) and artificial pheromone trails (Hunt et al. (2019)).

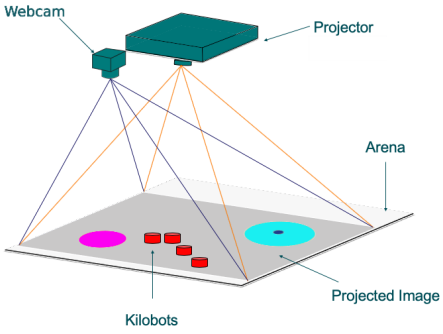


Figure 5: Kilobot arena setup

Robot Pose Determination Using the Matlab image processing toolbox, the centers of the robots are identified and logged. Image frames are acquired using the overhead camera snapshot function at the beginning of the tracking loop. The framerate was limited to approximately 1.7 fps by default. Low framerates did not prevent the setup from working – however, smoother and more precise control might be achieved with higher framerates. The acquired images were

converted from RGB colour space to grayscale and then further binarised using a hard-coded threshold in order to isolate the robots from the background. Next, a closing morphological operation was performed in order to eliminate internal noise caused by the robots' charging tab and other features, and a Gaussian smoothing filter was applied to increase regularity. Finally the robot positions were obtained using the Hough circle detection function to detect circular objects within certain radii bounds. The robot orientation is estimated from its movement between time steps. The projected light regions can then be updated according to the robot poses input into the various control algorithms.

Results

Simulation Results

Algorithm 1: Random Walk Avoidance Figure 6 shows 4 trajectories from 20 separate trial runs for illustrative purposes. A video of the Random Walk Avoidance algorithm in simulation with a single robot can be viewed at: `https://youtu.be/vy624ZeRXhY`.

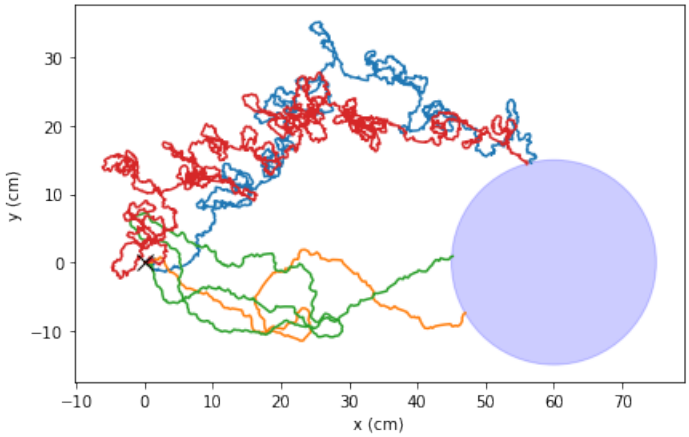


Figure 6: Algo 1 – Random Walk Avoidance (4 runs)

Algorithm 2: Dynamic Biased Random Walk Figure 7 shows 4 of 20 separate robot trials plotted. Compared to Algo 1, it shows a significant overall improvement in path linearity. This is explained by the significant bias in the random walk in the correct direction, introduced by the external user. While each of the four robots achieved the intended goal, it can be observed in the blue trajectory that notable looping occurred. This is as a result of a significant (simulated) intrinsic motor bias for a particular turning direction. This behavior is largely undesirable, though it could be exploited for applications where it may be beneficial to sweep out a larger area while maintaining overall directionality: perhaps for seeding a chemical trail.

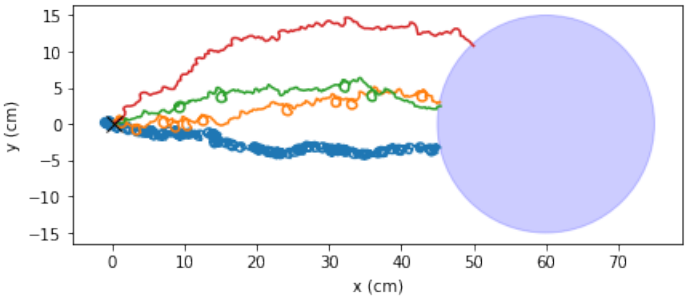


Figure 7: Algo 2 – Dynamic Biased Random Walk (4 runs)

A video of Algo 2 in simulation with a single robot can be viewed at: `https://youtu.be/nyJ-6HqKzjM`.

Algorithm 3: Left-Right Alternation In contrast with Algos 2 and 3, the Left-Right Alternation algorithm implements much tighter control and thus results in more direct trajectories. Figure 8 demonstrates this with a combined plot of 4 separate trial runs (of 20).

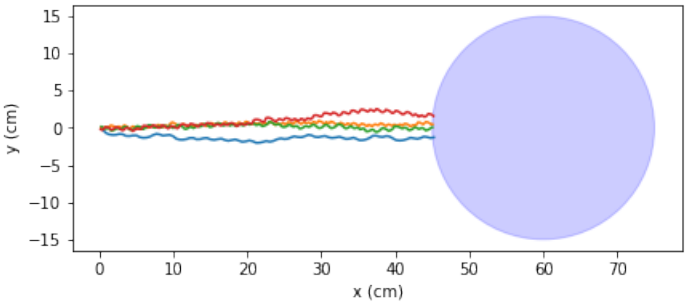


Figure 8: Algo 3 – Left-Right Alternation (4 runs)

A video of Algo 3 in simulation with a single robot can be viewed at: `https://youtu.be/9IlarChLe18`.

Statistical Algorithm Comparison The three algorithms were run on single robots 100 times each. Each run has a different initialization of intrinsic velocity biases, which are a substantial source of performance variation. Three performance metrics were examined: success rate, time to completion and path efficiency.

Success Rate A successful run is counted if the robot reaches the AOI within the maximum allocated time. This was set at 10,800 s initially (180 mins or 3 hours) and then 800 s (just over 13 mins). The success rate is shown in Table 1 for each algorithm.

Algorithm	800 s	10800 s
1 (Random Walk Avoidance)	0.05	0.91
2 (Dynamic Biased Random Walk)	0.96	0.97
3 (Left Right-Alternation)	0.94	0.95

Table 1: Algorithm Success Rate % (13 mins or 3 hours)

Time to Completion Figure 9 is a box plot comparing the time to completion of the three algorithms including failed runs.

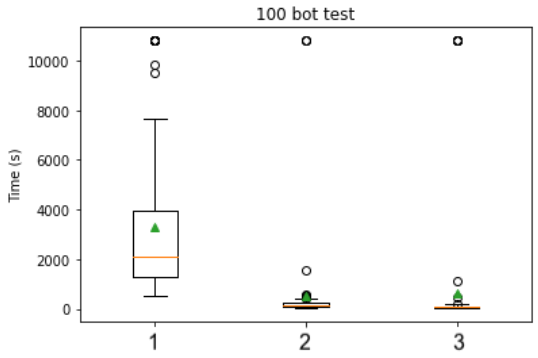


Figure 9: Initial Algo Time to Completion Comparison

Given the obvious difference in performance between Algo 1, and 2 and 3, an experiment cut-off time of 800 s was added to focus on a separate comparison between Algos 2 and 3, which commonly performed comfortably below this threshold (Figure 10).

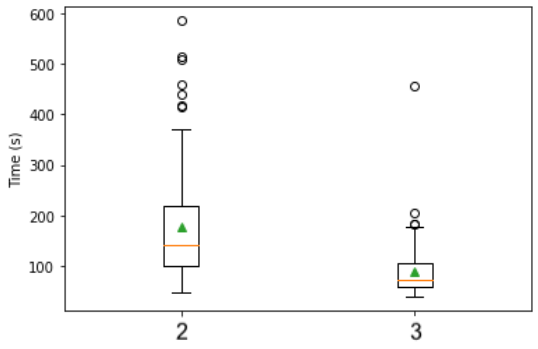


Figure 10: Time to Completion – Algos 2 and 3 (800 s experiment time out)

These results are further summarized in Table 2 with the data from Algo 1 included as well for comparison.

Algorithm	1	2	3
Mean (s)	666.64	176.92	88.29
Median (s)	697.05	141.54	72.03
Min (s)	549.05	48.06	40.06
Max (s)	781.06	585.01	455.02
Std. Dev. (s)	86.87	110.49	52.6

Table 2: Time to Completion Summary with a Time Threshold $T = 800s$

Path Efficiency The distance traveled is calculated by integrating the displacement over time while the Euclidean distance is measured from the starting coordinates to the finishing coordinates as illustrated in Figure 11. The efficiency of the path is measured as the ratio of actual path length taken to the Euclidean distance (i.e. a minimum of 1 for a straight line, unbounded above).

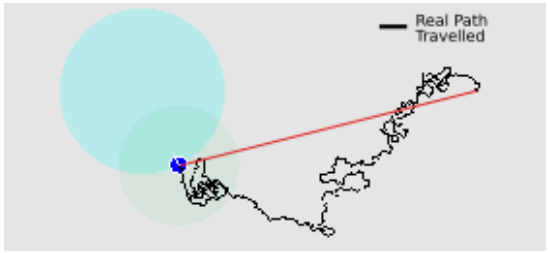


Figure 11: Euclidean Distance Versus Actual Path

Figure 12 and Table 3 compare the path efficiency for the three algorithms. Values closer to 1 indicate more efficient paths. Left-Right Alternation (Algo 3) scores lowest (best) for the four measures selected, the Dynamic Biased Random Walk algorithm (Algo 2) is next, and then finally the Random Walk Avoidance algorithm (Algo 1) has the highest ratio scores (worst efficiency).

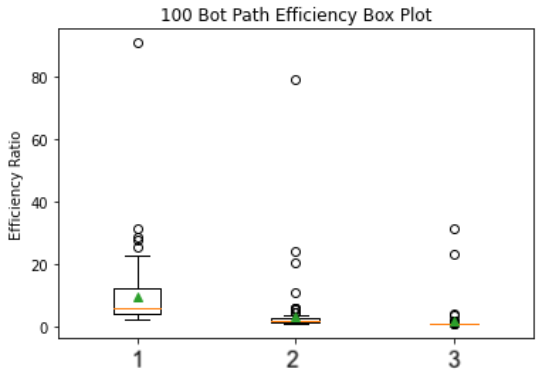


Figure 12: Control Algorithms' Path Efficiency Comparison

Algorithm	1	2	3
Mean	9.52	3.4	1.68
Median	6.07	1.78	1.06
Min	2.18	1.15	1.02
Max	90.77	79.3	31.62
Std. Dev.	10.48	8.26	3.76

Table 3: Path Efficiency Ratio Summary

Interference Behaviors An interesting emergent feature of Algo 3 (Left-Right Alternation) with multiple robots occurs when they are close enough to each other to enter their respective controlling regions. This results in two behaviors depending on whether the trailing robot approaches from the right or the left. In the first case when the approach is on the right as illustrated in Figure 13, the robot will follow the perimeter of the other robot until it reaches the AOI and stops at a distance equal to the controlling region radius.

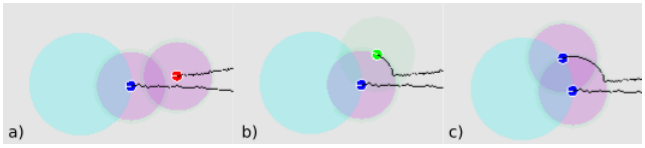


Figure 13: Algo 3: Right Approach Separation

In the second case when the approach is from the left as illustrated in Figure 14 the result is a looping motion which the robot will have difficulty escaping.

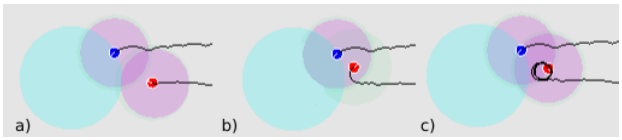


Figure 14: Algo 3: Left Approach and Spin

Figure 15 illustrates the first case with multiple robots which naturally maintain a distance between each other as they approach the AOI. Reducing the radius of the controlling regions will reduce the likelihood of interference as well as the final separation distance between the robots when interference occurs.

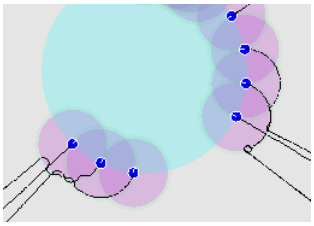


Figure 15: Algo 3: Robot Separation at Target

Waypoint Following

Using Algo 3 (the most effective control approach), simulations were carried out to perform waypoint following missions using single and multiple robots. The waypoint missions were implemented by defining an array of waypoint coordinates and incrementing through the array each time the robot entered the radius of the current waypoint. The precision of the trajectory is dependent on the radius of the AOI since the robot will stop at the perimeter. Figure 16 illustrates a waypoint mission with five waypoints and a single robot. Three AOI radii were used to demonstrate the difference in precision. The coordinates were generated algorithmically and are plotted in frame (a) and the resulting trajectories with progressively larger radii are illustrated in frames (b), (c) and (d).

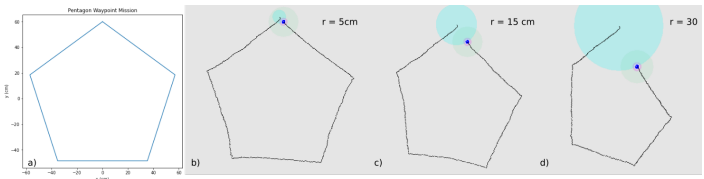


Figure 16: Single Robot Pentagon Waypoint Mission With Varying AOI Radius

With multiple robots, the current waypoint would only change if all robots had entered the current waypoint radius. This occasionally led to the robots not completing the waypoint mission as it only required one robot to fail to reach the AOI due to the spin mentioned in the *Interference Behaviors* subsection or a strong simulated turning bias for all the other robots to be prevented from progressing.

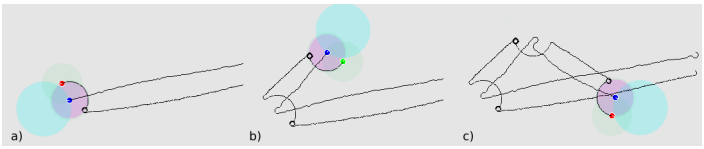


Figure 17: Algo 3 – Two Robots' Waypoint Trajectories

Figure 17 illustrates two robots being driven to three different waypoints one after the other. The robot depicted in blue was faster than the second robot so reached the AOI first and so was minimally affected by interference from the other controlling region. The second robot (red) was slower and so had to navigate the first robot's controlling region in order to reach the AOI. The second slower robot demonstrates both interference behaviors previously mentioned, as it repeatedly escaped the 'left approach and spin' behavior (Figure 17a,b,c) and followed the other robot's light projection perimeter (Figure 17c)

Depending on the application, there are a number of solutions to this. For example, assigning waypoints individu-

ally to each robot so that the faster ones can continue progressing which could result in large distances developing between robots especially on longer waypoint missions. Alternatively, if it is beneficial to maintain swarm cohesion, then the next waypoint can be updated only once a majority of the swarm have reached the AOI. This would allow the faster robots to continue while potentially freeing any robots stuck in a spin. This was implemented with five robots in a square waypoint mission where a minimum of three robots were required to reach each waypoint before the next waypoint was assigned – see Figure 18.

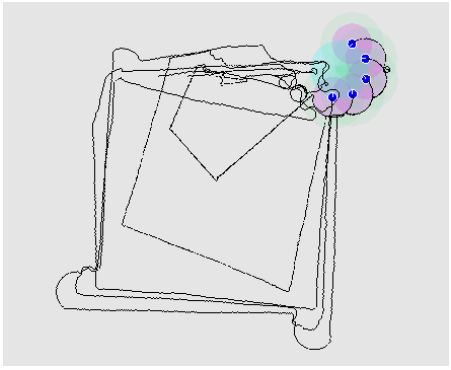


Figure 18: Algo 3 – Five Robot Square Waypoint Mission

As all of the robots were heading towards the same waypoint, this meant that slower robots would effectively cut the corners reducing the overall distance covered and time taken but at the expense of accuracy of trajectory. Figure 18 illustrates this with two robots visibly slower than the other three and therefore with less accurate trajectories.

A video of Algo 3 (Left-Right Alternation) with five robots in simulation performing a waypoint mission can be viewed at: `https://youtu.be/3TgL0GjS5GA`.

Real Robot Experiments

The practical experiments carried out were confronted by the influence of real-world physics. The robot motors were a challenge to calibrate correctly and movement was susceptible to small defects in the arena surface. Even so, it was possible to produce some interesting demonstrations of Algo 3 (Left-Right Alternation) functioning with both single robots and multiple robots, as well as a single robot waypoint missions. Algos 1 and 2 were not implemented with the real robots as they required a higher level of projection accuracy and bearing estimation than was possible with the current setup. Even then, the simulation results indicated that path efficiency for Algo 1 was likely to be relatively low and a test of battery life constraints.

Six Robot Herding As a first step it is interesting to herd the robots so that they form a cohesive swarm. From that point it would then be possible to drive them as a swarm to different positions in the arena. Figure 19 shows plots of the trajectories of six robots placed around the AOI depicted as the cyan circle. The robots were selected as those responding well to calibration and with reproducible movements. All robots successfully reached the AOI during the tests. Two of the robots, labeled Bot 1 and Bot 5 suffered from a directional bias resulting in spinning behavior.

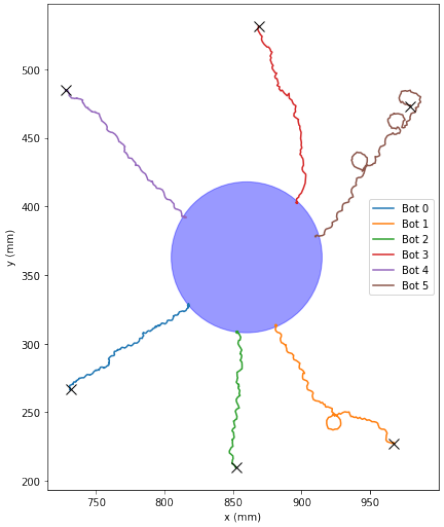


Figure 19: Algo 3 – Gathering Six Real Robots

Figure 20 shows six Kilobots being herded to the AOI. The images are taken from the webcam with the AOI and controlling regions overlaid in post-processing as the original frames had low contrast.

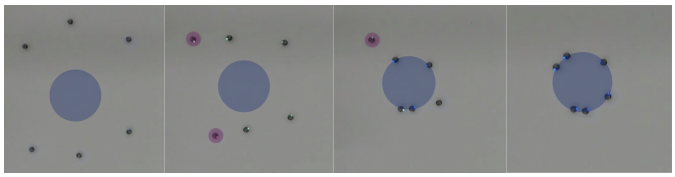
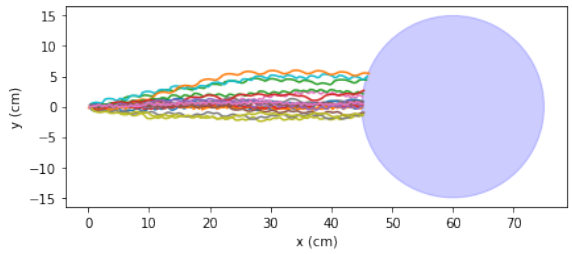


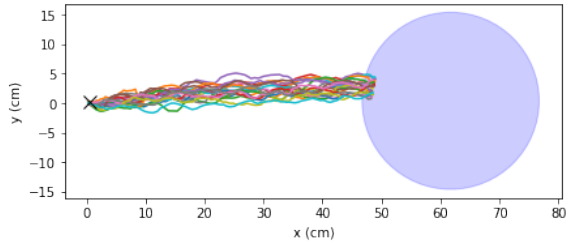
Figure 20: Algo 3 – Six Real Robots Gathering in Arena

A video of Algo 3 with six real robots can be viewed at: `https://youtu.be/pStKFwHMqUo`.

Simulation and Real Comparison Figure 21a is a plot of 20 runs performed in simulation from the same starting point using Algo 3. The same test was carried out with a real robot and the results can be seen in Figure 21b. Both sets of trajectories show only successful runs. Figure 21a shows the robots stopping at the edge of the AOI while Figure 21b shows the robots entering the circle before stopping. This is owing to the placement of the ambient light sensor at the rear of the robot while the robot coordinates are taken at the center of the robot.



(a) 20 Runs Algo 3 – Simulated



(b) 20 Runs Algo 3 – Real

Time to Completion Table 4 summarizes the time taken for the real and simulated robots to reach the AOI which had a radius of 15cm and was set at a distance of 60 cm from the robot starting point, meaning the robot had to travel approximately 45 cm. There was a significantly higher consistency in the real robot runs compared to the simulated runs. This can be explained by the experimenter repeatedly using one, well-calibrated real robot – chosen because it performed consistently. On the other hand, the simulated experiment was the equivalent of selecting 20 robots at random. Even though they all performed successfully, there was a much larger variance, as introduced by the simulated variety in motor bias and noise.

Algo 3	Real	Sim
Mean (s)	60.26	179.95
Median (s)	59.3	156.53
Min (s)	54.9	83.08
Max (s)	74.6	680.06
St. Dev. (s)	4.54	171.51

Table 4: Algo 3 – Real Versus Simulation Times (20 Trials)

Path Efficiency There was a greater consistency in path efficiency in the real robot runs, as confirmed by the much lower standard deviation than the simulated trials in Table 5.

Algo 3	Real	Sim
Mean	1.1	1.15
Median	1.1	1.09
Min	1.08	1.03
Max	1.17	2.04
Std. Dev.	0.02	0.25

Table 5: Algo 3 – Real Vs. Sim Path Efficiency (20 Trials)

Waypoint Following Figure 22 shows one real Kilobot robot starting at the spot marked with a black 'X' and navigating to each waypoint in turn before returning to the first waypoint. This demonstrates the ability of a light-based control method to drive the robots towards multiple sequential targets on more complex paths in order to perform a given task, such as collecting or depositing a resource. A video of a single real robot performing a waypoint mission can be viewed at: `https://youtu.be/NYkGvJmnns0`.

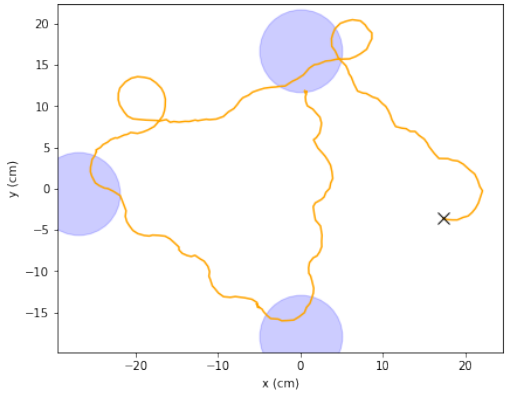


Figure 22: Example Real Robot Waypoint Path

Discussion

Given the simple reactive behaviors common to most microagents, there is likely to be a certain universality in light-beam algorithms that can usefully 'herd' such collectives around. Here, we have presented three light-beam control algorithms that successfully herd light-avoiding (phototophobic) agents around a two-dimensional environment. The first and second algorithms relied on (tunable) random walk avoidance behavior, while the third relied on switching between consistent leftward or rightward-biased forward movement. There is no difference in off-swarm computation requirements between the algorithms. The third algorithm was validated with real robots, where it was found to be robust to projection inaccuracies. Algorithm 3 exploits the fact that Kilobots can pivot forward on one of their legs while rotating left or right – in future work we will consider systematically the extent to which test algorithms are applicable to generic phototactic agents, or require 'co-design' with the target swarm's characteristics. We will also validate algorithms in real micro-environments, using the Dynamic Optical MicroEnvironment ('DOME') project (Denniss et al. (2022)). Future swarm engineers could refer to a common toolbox of broadly effective light-based swarm control algorithms, which can be chosen according to agent characteristics – whether microrobots or microorganisms.

Acknowledgments

E.R.H. acknowledges funding from the Royal Academy of Engineering and ESPRC (EP/N509619/1).

References

Alapan, Y., Yasa, O., Yigit, B., Yasa, I. C., Erkoc, P., and Sitti, M. (2019). Microrobotics and microorganisms: Biohybrid autonomous cellular robots. *Annual Review of Control, Robotics, and Autonomous Systems*, 2:205–230. doi: 10.1146/annurev-control-053018-023803.

Camazine, S., Deneubourg, J.-L., Franks, N. R., Sneyd, J., Theraulaz, G., and Bonabeau, E. (2003). *Self-Organization in Biological Systems*. Princeton University Press.

Chen, Y., Wang, Z., He, Y., Yoon, Y. J., Jung, J., Zhang, G., and Lin, Z. (2018). Light-enabled reversible self-assembly and tunable optical properties of stable hairy nanoparticles. *Proceedings of the National Academy of Sciences of the United States of America*, 115:E1391–E1400.

Dai, B., Wang, J., Xiong, Z., Zhan, X., Dai, W., Li, C.-C., Feng, S.-P., and Tang, J. (2016). Programmable artificial phototactic microswimmer. *Nature Nanotechnology*, 11:1087–1092.

Denniss, A. R., Gorochowski, T. E., and Hauert, S. (2022). An open platform for high-resolution light-based control of microscopic collectives. *Advanced Intelligent Systems*, 4:2200009.

Erkoc, P., Yasa, I. C., Ceylan, H., Yasa, O., Alapan, Y., and Sitti, M. (2019). Mobile microrobots for active therapeutic delivery. *Advanced Therapeutics*, 2:1800064.

Frangipane, G., Dell'Arciprete, D., Petracchini, S., Maggi, C., Saglimbeni, F., Bianchi, S., Vizsnyiczai, G., Bernardini, M. L., and Leonardo, R. D. (2018). Dynamic density shaping of photokinetic e. coli. *eLife*, 7:e36608.

Gorochowski, T. E., Hauert, S., Kreft, J.-U., Marucci, L., Stillman, N. R., Tang, T.-Y. D., Bandiera, L., Bartoli, V., Dixon, D. O. R., Fedorec, A. J. H., Fellermann, H., Fletcher, A. G., Foster, T., Giuggioli, L., Matyjaszkiewicz, A., McCormick, S., Olivas, S. M., Naylor, J., Denniss, A. R., and Ward, D. (2020). Toward engineering biosystems with emergent collective functions. *Frontiers in Bioengineering and Biotechnology*, 8:705.

Hauert, S. and Bhatia, S. N. (2014). Mechanisms of cooperation in cancer nanomedicine: towards systems nanotechnology. *Trends in Biotechnology*, 32(9):448–455. Special Issue: Next Generation Therapeutics.

Hunt, E. R., Jones, S., and Hauert, S. (2019). Testing the limits of pheromone stigmergy in high-density robot swarms. *Royal Society Open Science*, 6:190225.

Jones, S. (2019). Kilobox. `https://siteks@bitbucket.org/siteks/kilobox.git`.

Jones, S., Studley, M., Hauert, S., and Winfield, A. (2018). Evolving behaviour trees for swarm robotics.

Jékely, G., Colombelli, J., Hausen, H., Guy, K., Stelzer, E., Nédélec, F., and Arendt, D. (2008). Mechanism of phototaxis in marine zooplankton. *Nature*, 456:395–399.

Lavergne, F. A., Wendehenne, H., Bäuerle, T., and Bechinger, C. (2019). Group formation and cohesion of active particles with visual perception–dependent motility. *Science*, 364:70–74.

Lien, J.-M., Bayazit, O., Sowell, R., Rodriguez, S., and Amato, N. (2004). Shepherding behaviors. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings*, pages 4159–4164.

Long, N. K., Sammut, K., Sgarioto, D., Garratt, M., and Abbass, H. A. (2020). A comprehensive review of shepherding as a bio-inspired swarm-robotics guidance approach. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 4:523–537.

Palagi, S., Singh, D. P., and Fischer, P. (2019). Light-controlled micromotors and soft microrobots. *Advanced Optical Materials*, 7:1900370.

Purcell, E. B. and Crosson, S. (2008). Photoregulation in prokaryotes. *Current Opinion in Microbiology*, 11:168–178.

Rubenstein, M., Ahler, C., and Nagpal, R. (2012). Kilobot: A low cost scalable robot system for collective behaviors. In *IEEE International Conference on Robotics and Automation, 2012*, pages 3293–3298.

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

Ruskowitz, E. R. and DeForest, C. A. (2018). Photoresponsive biomaterials for targeted drug delivery and 4d cell culture. *Nature Reviews Materials*, 3:17087.

Schmidt, F., Liebchen, B., Löwen, H., and Volpe, G. (2019). Light-controlled assembly of active colloidal molecules. *The Journal of Chemical Physics*, 150:094905.

Slavkov, I., Carrillo-Zapata, D., Carranza, N., Diego, X., Jansson, F., Kaandorp, J., Hauert, S., and Sharpe, J. (2018). Morphogenesis in robot swarms. *Science Robotics*, 3:eaau9178.

Steager, E. B., Wong, D., Chodosh, N., and Kumar, V. (2015). Optically addressing microscopic bioactuators for real-time control. pages 3519–3524.

Stoychev, G., Kirillova, A., and Ionov, L. (2019). Light-responsive shape-changing polymers. *Advanced Optical Materials*, 7:1900067.

Tao, Y., Chan, H. F., Shi, B., Li, M., and Leong, K. W. (2020). Light: A magical tool for controlled drug delivery. *Advanced Functional Materials*, 30:2005029.

Vo, C., Harrison, J. F., and Lien, J.-M. (2009). Behavior-based motion planning for group control. In *IEEE International Conference on Intelligent Robots and Systems, 2009*, pages 3768–3773.