Evolving Coverage Behaviours For MAVs Using NEAT

Extended Abstract

James Butterworth
University of Liverpool
Liverpool, UK
J.Butterworth2@liverpool.ac.uk

Bastian Broecker
University of Liverpool
Liverpool, UK
Bastian.Broecker@liverpool.ac.uk

Karl Tuyls
University of Liverpool & Google DeepMind
Liverpool, UK
k.tuyls@liverpool.ac.uk

Paolo Paoletti
University of Liverpool
Liverpool, UK
P.Paoletti@liverpool.ac.uk

ABSTRACT

Dynamic coverage - the problem of covering an area evenly and continuously in order to visit all areas of interest - is an important procedure to optimise for any autonomous surveillance system. This work introduces a novel solution to the multi-agent version of this problem in that it achieves high performance in a completely decentralised manner with no reliance on GPS. It does so by using NEAT [12] to optimise agent neural controllers. The controllers are first realised via simulation and then transferred to Micro-Aerial Vehicles (MAVs). The MAVs are modified to include a Ultra-wideband Frequency (UWB) chip which use radio waves to communicate inter drone distances to one another.

KEYWORDS

neuroevolution; NEAT; micro-aerial vehicles; dynamic coverage

ACM Reference Format:
James Butterworth, Bastian Broecker, Karl Tuyls, and Paolo Paoletti. 2018. Evolving Coverage Behaviours For MAVs Using NEAT. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 3 pages.

1 INTRODUCTION

There exists a wide array of potential applications that a multi-agent robotic solution for surveillance could help solve. Examples include: the protection of safety critical technical infrastructures, the safeguarding of country borders, and the monitoring of high-risk regions and danger zones which cannot be entered by humans in the case of a nuclear incident, a biohazard or a military conflict. A *multi*-agent system also has the additional benefits of being, robust to individual failure, scalable to different sized environments and can solve the problem in parallel, thereby greatly reducing the time it takes to carry out the procedure.

There are a number of problems associated with deploying a robotic swarm in hazardous environments: the potential for communication links to be disrupted or severed, and that GPS may not always be available, which would be the case for robots working in radiation proof environments or areas where natural disasters have recently occurred. It is therefore important that a solution is completely decentralised and that the system is not reliant on GPS.

In the area of dynamic coverage, previous works [1, 9] take a centralised approach in order to achieve excellent results in simulated environments. Furthermore, in [6], NEAT is also used to evolve the controllers for aquatic robots performing area coverage; this system is full decentralised however it requires the use of GPS on the real robotic system. Biologically inspired systems have also been considered [2, 3, 7, 11, 13] which take inspiration from the way ants attract other members of the swarm to the location of food sources by creating paths of pheromones. By using simulated pheromone trails in order to deter other individuals from areas that have already been explored, efficient dynamic coverage algorithms can be designed. Even though these algorithms achieve impressive results, the mechanisms of pheromone secretion and evaporation is very hard to achieve on real robots, especially MAVs.

Given that each of the works discussed so far, individually, do not adhere to the constraints of uncertain and hazardous environments that have been outlined, we propose the following: a decentralised, multi-agent system to perform dynamic coverage; the system has no reliance on GPS and can be implemented on a real swarm of MAVs.

2 METHOD

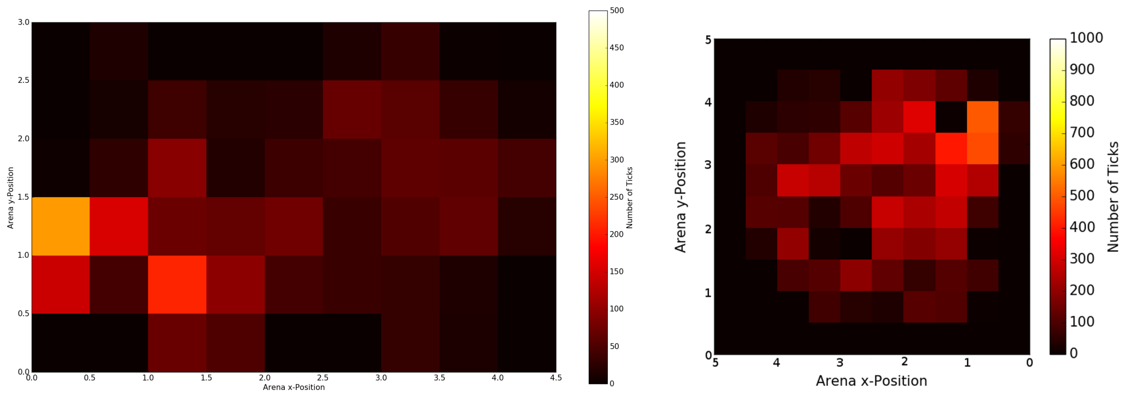
A neural network is used to control the agent which is trained via a genetic algorithm. In particular, the algorithm NEAT [12] is used whereby neural controllers with different topologies can be evolved, due to this the controllers are often smaller and more efficient than fixed topology networks. For an in depth description of the NEAT algorithm, we refer the reader to [12].

The NEAT neural controllers are first evolved in the ARGoS simulation environment [10]. A model of a generic mini quadcopter was modified to be an accurate representation of the robot that will be used in reality - the Crazyflie 2.0. The decision was made to restrict the agents to the same altitude as each other due to the fact that a powerful down wind known as ‘prop wash’ can cause MAVs caught in it to destabilise and crash. This greatly simplifies the procedure without restricting much of the functionality if coverage performance is measured over a 2D plane.

The neural controllers have 3 inputs: a bias input which is always 1, the distance to the closest drone and the distance to the closest wall. The two outputs of the network are left and right ‘wheel

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

Figure 1: Heatmaps showing the number of ticks that each *cell* was occupied. *Left:* This heat map represents 2 real Crazflies performing coverage over a 4.5m by 3m sized arena for 120 seconds. *Right:* This heat map represents a 3 agent run in a 5m by 5m sized simulated arena for 300 seconds.



speeds' which are then mapped down to speeds in the x and y directions.

The Crazyflie models were evaluated in a 5m by 5m square arena each containing 3 agents (with a copy of the same network) for 300 simulated seconds. This arena was split into 100 *cells* (0.5m by 0.5m each) which were used as a measurement tool to evaluate how well coverage was achieved. Each agent was placed near the centre such that all the agents started out in a circle; each agent was initiated with a random orientation for each evolutionary run. Each network is evaluated 5 times and the lowest score of those 5 trials is taken as the fitness for that organism. The GA is ran for 50 generations and after that time the best performing individual over all of the generations is reevaluated and observed. A mutation rate of 10%, a population size of 100 and a mutation power of 1.8 were used for NEAT hyperparameters (a full list of parameters are available on request). The fitness function designed for area monitoring in [6] was used in order reward behaviour that lead to *each* cell being visited frequently.

In order to evolve a controller for use on the real Crazyflies, the arena size used during evolution was reduced to 4.5m by 3m (the size of the arena in the lab) and the amount of time in simulation was reduced from 300 seconds to 120 due to the limited battery life on the Crazyflies. The real robotic platform used in the experiments is the Crazyflie 2.0. The quadcopter controller is imported from the best performing individual from 50 generations of the NEAT GA. The distance to the closest Crazyflie is retrieved from the UWB chip and the distance to the closest wall is calculated using the current pose of the Crazyflie provided by a motion capture system.

3 RESULTS & CONCLUSIONS

NEAT successfully evolved coverage behaviour on 3 drones that avoided crashes in all 5 tests. The controller was tested with 3 drones in an arena size of 5m by 5m. Figure 1 (*Right*) is a heat map representing one of these runs where each cell has a value which represents the number of ticks that the cell is occupied. This simulation was ran for 300 seconds at a rate of 10 ticks per second. It is apparent from this image that the drones performed well at coverage given that there is a relatively similar shade of red across most of the map. This implies that most of the cells were occupied during the run and that they were all occupied for a similar amount of time. 90% of the cells have been visited for at least 1 tick (this is not including the outer border of cells as the agents often avoid this area in order to not crash into the walls) for this run. In order to evaluate the average performance of the individual, the controller was tested for 1000 runs with the same metric as above being taken for each run and averaged over all the performances. This results in a sample mean of: 76.9% and a sample standard deviation of: 10.0%.

In order to demonstrate the *scalability* of the solution, the same controller was evaluated using 6 drones in a 7m by 7m arena. Evaluating the same controller that was evolved on the 5m by 5m arena leads to a sample mean of: 77.2% and a sample standard deviation of: 7.0% of the cells being visited at least once. Finally, to test the *robustness* of the solution with respect to individual failure, the experiment was repeated with 5 drones on an arena size of 7m by 7m in order to simulate the loss of one agent. This experiment resulted in a sample mean of: 73.8% and a sample standard deviation of: 7.3% over 1000 runs. A video showing an individual run and respective heat map of all these experiments is available at: https://www.youtube.com/watch?v=ifTKCkG98Z0

The best controller from 50 generations of the more realistic simulation was transferred to the real Crazyflie and then tested for a singular run whilst recording the performance in the form of a heat map - the recorded heat map is shown in Figure 1 (*Left*). The real Crazyflies performed coverage very well, according to the metric used so far, 100% coverage was achieved. Moreover, a very even spread of coverage was achieved leading to a similar shade of red across the entire map. A video of the realistic simulation and the real Crazyflie experiment can be found at:
https://www.youtube.com/watch?v=sEs6doz_LYM&t=7s

To conclude, this paper introduces a method to evolve a neural controller that performs dynamic coverage in simulation and then transfers this optimised controller to a real swarm of MAVs. In contrast to previous works, the system was fully decentralised and had no reliance on GPS. In future work, we will consider using Optical Flow techniques [4, 5, 8] to detect objects in the environment.

REFERENCES

[1] Gökhan M. Atinç, Dušan M. Stipanović, and Petros G. Voulgaris. 2014. Supervised coverage control of multi-agent systems. *Automatica* 50, 11 (nov 2014), 2936–2942.

[2] Bastian Broecker, Ipek Caliskanelli, Karl Tuyls, Elizabeth Sklar, and Daniel Hennes. 2015. Social Insect-Inspired Multi-Robot Coverage. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS '15)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1775–1776.

[3] Ipek Caliskanelli, Bastian Broecker, and Karl Tuyls. 2015. Multi-Robot Coverage: A Bee Pheromone Signalling Approach. In *Artificial Life and Intelligent Agents: First International Symposium, ALIA 2014, Bangor, UK, November 5-6, 2014. Revised Selected Papers*, J Christopher Headleand, J William Teahan, and Llyr Ap Cenydd (Eds.). Springer International Publishing, 124–140.

[4] G D Croon, D Alazard, and D Izzo. 2015. Controlling spacecraft landings with constantly and exponentially decreasing time-to-contact. *IEEE Trans. Aerospace Electron. Systems* 51, 2 (2015), 1241–1252. https://doi.org/10.1109/TAES.2014.130135

[5] de Croon G.C.H.E., Ho H.W., De Wagter C., van Kampen E., Remes B., and Q P Chu. 2013. Optic-flow based slope estimation for autonomous landing. *International Journal of Micro Air Vehicles* 5, 4 (2013), 287–297.

[6] Miguel Duarte, Vasco Costa, Jorge Gomes, Tiago Rodrigues, Fernando Silva, Sancho Moura Oliveira, and Anders Lyhne Christensen. 2016. Evolution of Collective Behaviors for a Real Swarm of Aquatic Surface Robots. *PLOS ONE* 11, 3 (2016), 1–25.

[7] Sven Koenig and Yaxin Liu. 2001. Terrain Coverage with Ant Robots: A Simulation Study. In *Proceedings of the Fifth International Conference on Autonomous Agents (AGENTS '01)*. ACM, New York, NY, USA, 600–607.

[8] Kimberly McGuire, Guido C H E de Croon, Christophe De Wagter, Bart Remes, Karl Tuyls, and Hilbert J Kappen. 2016. Local Histogram Matching for Efficient Optical Flow Computation Applied to Velocity Estimation on Pocket Drones. *CoRR* abs/1603.0 (2016). http://arxiv.org/abs/1603.07644

[9] A Mohammadi, F Abdollahi, and M Naraghi. 2014. Coverage control of a dynamic large-scale environment using sector-based sensors. In *2014 Second RSI/ISM International Conference on Robotics and Mechatronics (ICRoM)*. 388–392. https://doi.org/10.1109/ICRoM.2014.6990932

[10] C Pinciroli, V Trianni, R O'Grady, G Pini, A Brutschy, M Brambilla, N Mathews, E Ferrante, G A Di Caro, F Ducatelle, M Birattari, L M Gambardella, and M Dorigo. 2012. {ARGoS}: A Modular, Parallel, Multi-Engine Simulator for Multi-Robot Systems. *Swarm Intelligence* 6, 4 (2012), 271–295.

[11] Bijan Ranjbar-Sahraei, Gerhard Weiss, and Ali Nakisaee. 2012. A Multi-robot Coverage Approach Based on Stigmergic Communication. In *Multiagent System Technologies: 10th German Conference, MATES 2012, Trier, Germany, October 10-12, 2012. Proceedings*, Ingo J Timm and Christian Guttmann (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 126–138.

[12] Kenneth O Stanley and Risto Miikkulainen. 2002. Evolving Neural Networks Through Augmenting Topologies. *Evolutionary Computation* 10, 2 (2002), 99–127.

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