https://doi.org/10.31256/Yp7Gw5I

Evolving complex terrain navigation: Emergent contour following from a low-resolution sensor

Dexter R. Shepherd*, James C. Knight*
* Department of Informatics, University of Sussex, Brighton, United Kingdom
Email: drs25@sussex.ac.uk

***Abstract*—This paper investigates evolutionary approaches to enable robotic agents to learn strategies for energy-efficient navigation through complex terrain, consisting of water and different heights. Agents, equipped with a low-resolution depth sensor, must learn how to navigate between a randomly chosen start/end position in a procedurally generated world, along a path which minimises energy usage. The solution that consistently emerged, was an agent that followed the contours of the map, resulting in near-optimal performance in little evolutionary time. Further, initial experiments with a real robot and Kinect sensor showed that the simulated model successfully predicted the correct movement that would be needed to follow contours. This demonstrated both that the evolved strategies are robust to noise and capable of crossing the reality gap. We suggest that this robustness is due to the use of a low-resolution sensor.**

***Index Terms*—Robotics, low-resolution, evolution, wheg**

I. Introduction

Nearly all organisms are limited at some point when it comes to movement. For example, an octopus can fit through any aperture but only if it is bigger than its beak [1]. These environmental barriers constrain an organism's ability to traverse terrain. When a path looks too arduous, organisms will often choose an easier route. For an autonomous robot travelling through complex terrain, the most energy-efficient path might not be the one that is most direct, but instead one that avoids unnecessary climbing.

While path planning algorithms can solve this problem [2], they require an accurate terrain model of the world. If there is no map, or the world is dynamic, the path must be decided using local information. Here, we take this approach and use a genetic algorithm to evolve a controller for a simulated robot equipped with a claw-style wheel-leg hybrid known as whegs and a Kinect sensor to enable it to navigate within an environment consisting of rocks and flatland. The agent is set a target direction and had to reach it while avoiding water and steep height within the environment. Success was defined as reaching the goal while expending as little energy as possible. If the agent gets stuck it will receive no reward and if it makes unnecessary movements the amount of reward it received will be reduced. The agent is controlled by a two layer neural network, which receives inputs from a low-resolution depth sensor and produces outputs representing the desired direction of movement. Following a straight path to the goal is the obvious solution, but even with such low resolution and a small network, agents that perform contour following were consistently evolved through trials. Moreover, rather than being a problem for evolution, we suggest that the low-resolution sensor enables robust behaviour in the face of noise and crucially allows the successful transfer of the solution to the real robot.

II. Methods

A. Simulated environment

Our agent simulates the whegged robot shown in figure 1 with information about the current terrain in the direction of travel provided by a low-resolution depth sensor. All trials were performed in 3D terrains – similar to those shown in figure 2 – generated using ten octaves of Perlin noise [3]. In each trial, a start position and end goal were placed at random locations in the environment, 10 units apart. To add additional complexity, we defined any regions below a set height as water, which acted as an in-penetrable obstacle. A first person depth view was generated from the terrain by casting rays outwards from the agent at 20° intervals across a field-of-view of 100° in azimuth, at 5 different height levels. This generates a 5×5 image of the terrain within 5 units of the front of the robot. We used a depth sensor because it would detect structure without disturbance from variations of light levels. We hoped that using such a low resolution image will not only reduce the computation required to train and use the model but might also increase robustness.

B. Neural network

Our network architecture consisted of one 2D convolutional layer with 3×3 filters. Its output was flattened into a linear layer which also had two additional input nodes providing the direction vector to the end goal, supplying the model with the direction the agent *should* be travelling. Finally, the linear layer was densely connected to an output layer with 8 nodes, representing movements in worldspace that the agent could take. Each timestep, the agent moves in the direction specified by the output node with the largest value chosen as the next direction and a step is moved in that direction, with the new angle facing set.

C. Evolutionary training

We optimised the neural network using the Microbial genetic algorithm [4] and a fitness function $((100 - energy) \times 0.003 + (10 - endDist) \times 0.007) \times 100$ such that minimising distance is worth 70% of the fitness, and minimising energy 30%. Additionally, if the agent collides with water or ends up

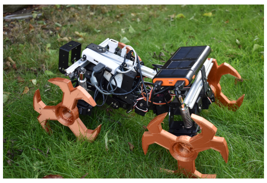

Fig. 1. Robotic chassis used for physical trials.

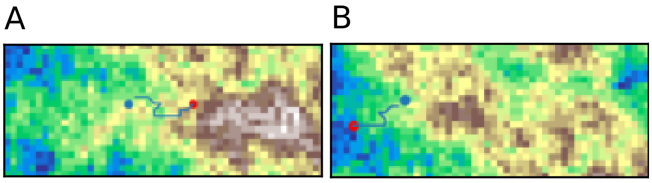

Fig. 2. Example trials in simulated environment. Red and blue circles show agent start and end positions respectively and lines show path taken.

further away from the target than it started, fitness is set to 0. More energy is used moving up hill than down hill, or on solid ground and energy usage is accumulated over every step.

D. *Physical robot*

The robot chassis shown in figure 1 was driven by continuous rotation servos, attached to claw-style wheels known as Whegs, inspired by cockroach locomotion [5]. In order to match the simulation, we used an Xbox Kinect time-of-flight sensor. We cropped out the central 200×400 pixels of the depth image and downsampled this to 5×5 to match the field-of-view and input resolution of our simulated agent.

III. Results

We evolved 20 agents for 500 generations and evaluated their performance by averaging the fitness across three trials with random start/end positions. The mean fitness of successful agents was 89% and this was always achieved in <100 generations. Figure 2 shows typical behaviour and it is interesting to see that contour-following has emerged – a strategy which avoids energy loss *and* water.

To see if our trained models are robust enough to cross the reality gap, we tested them on real-world data recorded from the Kinect sensor. As figure 3A-C illustrates, our preprocessing successfully extract a low-resolution depth map of the scene. The real world depth imaging was found through qualitative inspection of the data to not always be as clear as the simulated depth imaging. Real world noise could mistake distance. However, despite this low resolution image and real world noise, the model produces predominantly the correct directions. In this instance, Figure 3D shows the agent should move left, which is away from an immediate clash with the rocks. Indeed, in the majority of trials, the robot picked a direction away from complex terrain to an accuracy of 74% (as judged by qualitatively marking expected predictions against incorrect predictions) suggesting that the solution will cross the reality gap.

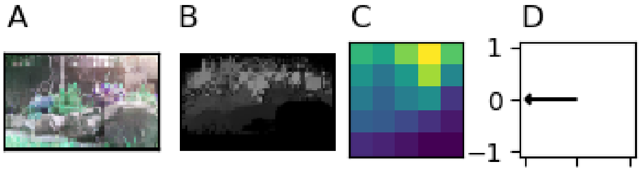

Fig. 3. (**A**) RGB (**B**) depth image from Kinect. (**C**) Depth image downsampled to 5×5 (yellow=far, black=near). (**D**) Movement direction from model.

IV. Conclusion

In this paper, we have shown that agents using a low-resolution depth sensor can successfully avoid collision with environmental hazards and find paths across complex terrain that minimise energy usage through emergent contour following. Using such a low resolution depth image not only reduced the computation required to train and use the model but – as has been demonstrated in insect-inspired visual navigation research [6] – improves robustness to noise by only retaining low-frequency information. We demonstrated this by taking models evolved in simulation across the reality gap and showing that they were able to avoid obstacles in the real world. The success of this approach suggests that low-resolution sensing might be an an interesting alternative to other more complex Sim2real approaches [7] and we also believe is key to how insects are able to navigate robustly despite very large restrictions on energy and neural resources.

Acknowledgements

JK was funded by the EPSRC (grant EP/V052241/1).

References

[1] J. B. Wood and R. C. Anderson, "Interspecific evaluation of octopus escape behavior," *Journal of Applied Animal Welfare Science*, vol. 7, no. 2, pp. 95–106, 2004.

[2] D. Delling, P. Sanders, D. Schultes, and D. Wagner, "Engineering route planning algorithms," in *Algorithmics of large and complex networks*. Springer, 2009, pp. 117–139.

[3] K. Perlin, "Improving noise," in *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '02. New York, NY, USA: Association for Computing Machinery, 2002, p. 681–682. [Online]. Available: https://doi.org/10.1145/566570.566636

[4] I. Harvey, "The microbial genetic algorithm," in *European conference on artificial life*. Springer, 2009, pp. 126–133.

[5] R. Schroer, M. Boggess, R. Bachmann, R. Quinn, and R. Ritzmann, "Comparing cockroach and whegs robot body motions," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 4, 2004, pp. 3288–3293 Vol.4.

[6] A. Wystrach, A. Dewar, A. Philippides, and P. Graham, "How do field of view and resolution affect the information content of panoramic scenes for visual navigation? A computational investigation," *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, vol. 202, no. 2, pp. 87–95, 2016.

[7] A. Kadian, J. Truong, A. Gokaslan, A. Clegg, E. Wijmans, S. Lee, M. Savva, S. Chernova, and D. Batra, "Sim2real predictivity: Does evaluation in simulation predict real-world performance?" *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6670–6677, 2020.