Towards Generalization in Target-Driven Visual Navigation by Using Deep Reinforcement Learning

1st Alessandro Devo  
Department of Engineering  
University of Perugia  
Perugia, Italy  
alessandro.devo@studenti.unipg.it

2nd Giacomo Mezzetti  
Department of Engineering  
University of Perugia  
Perugia, Italy  
giacomomezzet@gmail.com

3rd Gabriele Costante  
Department of Engineering  
University of Perugia  
Perugia, Italy  
gabriele.costante@unipg.it

4th Paolo Valigi  
Department of Engineering  
University of Perugia  
Perugia, Italy  
paolo.valigi@unipg.it

Abstract—In this work, we address generalization in target-driven visual navigation by proposing a novel architecture composed by two networks, both exclusively trained in simulation. The first one has the objective of exploring the environment, while the other one of locating the target. We test our agent in both simulated and real scenarios, and validate its capabilities through extensive experiments with previously unseen goals and unknown mazes, even much larger than the ones used for training.

Index Terms—Robot Navigation, Learning for Robotics, Computer Vision for Robotics

I. INTRODUCTION

Target-driven visual navigation is a longstanding goal in the robotics community (Fig. 1). A naive way to approach this problem is to combine a classic navigation system with an object detection module. However, map-based approaches [1] assume the availability of a global map of the environment, while SLAM algorithms [2] are not still specifically designed to be paired with such object detectors.

For these reasons, map-less methods [3], [4] have proven to be much more suitable for this task. One of the main approaches is to combine Deep Neural Networks (DNNs) and Reinforcement Learning (RL), which together allow to manage the relationship between vision and motion in a natural way [5]. However, current methods are limited to consider as goals specific scenes or objects with which the model is trained [6], [7]. Therefore, in practice, it is still necessary to train, or at least fine-tune, the agent for every new object and environment. In real scenarios, this is not only an expensive approach, but it can also be dangerous.

To avoid that, we design a novel RL based architecture composed by two main DNNs: the first, the navigation network, with the goal of exploring the environment and approaching the target; the second, the object localization network, with the aim of recognising the specified target in the robot’s view. They are both exclusively trained in synthetic environments, however, we show that: i) our algorithm directly transfers to new unknown scenarios, even much larger than the ones used during training, and most importantly, ii) also to real ones with real targets.

II. APPROACH

The target-driven visual navigation problem consists in finding the shortest sequence of actions to reach a specified target, using only visual inputs. Our goal is to design an agent able to find that sequence directly from pixels.

We consider the standard RL setting where the agent interacts with the environment over a number of discrete time steps. The observation is composed by the current RGB frame from the agent point of view and the image of the target to be reached. These two inputs are both fed into the architecture, which consists of two different networks. The first, i.e. the object localization network, has the objective of comparing the two images and locate the target. The second, i.e. the navigation network, is used to learn exploration strategies to solve complex mazes.

The overall training is divided in two completely independent phases, one for each network. The object localization network training is posed as a similarity metric learning problem. For this purpose, we collect a synthetic dataset, whose samples consist of triplets of images, each containing: the picture of the goal, an image in which the goal is visible and another one in which it is not. The navigation network is trained via RL using IMPALA [8], which, leveraging both CPUs and GPUs, allows fast and sample-efficient off-policy learning. To speed up training, we make use of auxiliary depth prediction [5] and an experience replay [9], which is a buffer of trajectories shared among actors.

To enhance the generalization capabilities of the algorithm and allow a direct sim-to-real transfer, we employ domain randomization [10], which has been successfully applied in other complex robotic tasks [11].
III. EXPERIMENTS AND RESULTS

A. Simulated Experiments

To measure our system performance in unseen simulated environments, we make two types of tests.

1) Exploration Experiment: In this experiment, we place the agent in the center of a $20 \times 20$ maze, which is much larger than the $3 \times 3$ mazes in which it is trained. We give it 180 seconds to explore it, and at the end of the episode, we measure the percentage of the maze it has discovered.

We evaluate our agent performance in 4 different mazes, using 4 different levels of light intensity and 3 random floor and wall textures. For each of the 48 possible combinations, we average the results over 3 runs. In Table I, the scores are reported by percentage of explored area, w.r.t. the light intensity, in comparison with human model.

2) Target-driven Experiment: In this second experiment, we place our agent in a $5 \times 5$ maze, which ends in a room with 3 different objects, including the target. An episode ends when the agent reaches the target or when 90 seconds are elapsed.

We try with 9 different objects, averaged over 6 runs each. The averaged results are expressed in seconds and summarised in Table II, in comparison to human performance.

B. Real Experiments

To test our model performance in real settings, we build 7 different $4 \times 4$ mazes, both indoor and outdoor. As for the experiments in simulation, we test both the exploration and the target detection capabilities of our model. In real settings, we run a total of 84 experiments, in which we measure our agent performance with three different types of tests:

1) The robot goal is to reach the target as fast as possible. The run ends when the robot approaches the object or after 1000 steps (Table III);
2) The robot objective is just to explore the maze as much as possible. In this case, the episode ends when the maximum number of 1000 steps is reached (Table IV);
3) The robot is placed inside the maze, together with three objects, including the target. The run ends when the robot reaches the target or after 1000 steps (Table V).

IV. CONCLUSIONS

In this work, we introduced a new framework for target-driven visual navigation. Through extensive experimentation, in both synthetic and real mazes, we showed that a direct sim-to-real transfer for this task is possible. The proposed model, indeed, not only proved capable of navigating in much larger mazes than those in which it was trained, but also showed a good ability to generalize in real ones.

REFERENCES

[1] J. Borenstein and Y. Koren, “Real-time obstacle avoidance for fast mobile robots,” IEEE Transactions on systems, Man, and Cybernetics, vol. 19, no. 5, pp. 1179–1187, 1989.

[2] R. Mur-Artal and J. D. Tardos, “Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras,” IEEE Transactions on Robotics, vol. 33, no. 5, pp. 1255–1262, 2017.

[3] G. Kahn, A. Villaflor, B. Ding, P. Abbeel, and S. Levine, “Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation,” in 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018, pp. 1–8.

[4] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, “Cognitive mapping and planning for visual navigation,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2616–2625.

[5] P. Mirowski, R. Pascaru, F. Viola, H. Soyer, A. J. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu et al., “Learning to navigate in complex environments,” arXiv preprint arXiv:1611.03673, 2016.

[6] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, “Target-driven visual navigation in indoor scenes using deep reinforcement learning,” in 2017 IEEE international conference on robotics and automation (ICRA). IEEE, 2017, pp. 3357–3364.

[7] X. Ye, Z. Lin, H. L. Su, Z. Zheng, and Y. Yang, “Active object perceiver: Recognition-guided policy learning for object searching on mobile robots,” in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018, pp. 6857–6863.

[8] L. Espelt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning et al., “Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures,” arXiv preprint arXiv:1802.01561, 2018.

[9] Z. Wang, V. Bapst, N. Heess, V. Mnih, R. Munos, K. Kavukcuoglu, and N. de Freitas, “Sample efficient actor-critic with experience replay,” arXiv preprint arXiv:1611.01224, 2016.

[10] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, “Domain randomization for transferring deep neural networks from simulation to the real world,” in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017, pp. 23–30.

[11] M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray et al., “Learning dexterous in-hand manipulation,” arXiv preprint arXiv:1808.00177, 2018.