2020 I-RIM Conference
December 10-12, 2020, Rome, Italy (Virtual)
ISBN: 9788894580518
DOI:10.5281/zenodo.4781281

Deep Reinforcement Learning for Visual Navigation and Active Tracking

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

Alberto Dionigi
Department of Engineering
University of Perugia
Perugia, Italy
alberto.dionigi@studenti.unipg.it

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

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

***Abstract*—In this contribution, we present our research line on Deep Reinforcement Learning approaches for robot navigation, in particular: Target-Driven Visual Navigation and Visual Active Tracking. We assess our methods capabilities in several challenging scenarios and in a number of environments previously unseen during training. Finally, we also prove that they can be effectively deployed in real-world settings on real platforms.**

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

I. INTRODUCTION

A. Target Driven Visual Navigation

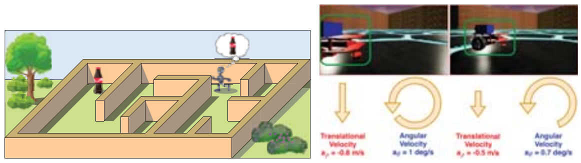
TARGET-DRIVEN visual navigation (TDVN) is a long-standing goal in the robotics community (Fig. 1a). A naive way to approach this problem is to combine a classic navigation system with an object detection module. However, map-based approaches [1] assume the availability of a global map of the environment,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

For these reasons, map-less methods [3], [4] have proven to be much more suitable for this task. A widespread approach is to learn complex control policies by using Deep Reinforcement Learning (DRL), which is also successfully applied in many other control applications [5]. Unfortunately, current methods are limited to consider as goals specific scenes or objects with which the model is trained [3], [4]. Therefore, in practice, it is still necessary to train, or at least fine-tune, the agent for every new object and environment.

To avoid that, we design a novel DRL based architecture composed by two main networks: 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 exclusively trained in simulation, and no single real image is used. Finally, we show that our algorithm directly transfers to new unknown environments, even much larger than the ones used during training, and, most importantly, also to real ones with real targets.

B. Visual Active Tracking

Most of the works in Visual Tracking (VT) focus exclusively on tracking objects and/or people in pre-recorded videos [6] or, in general, assume that the target is always within the field of view of a fixed camera, whose position cannot be adapted to the target motion.



(a) (b)

Fig. 1. (a) The target-driven visual navigation task. The agent has to explore an environment to reach a user specified target. The only inputs it receives are the visual frames from its first person view and an image of the goal. (b) The visual active tracking task. The goal of the tracker is to maintain the target (marked in green) within its field of view by performing motion maneuvers.

On the contrary, we focus on the more challenging task of Visual Active Tracking (VAT) [7], [8], in which the tracker has to *actively* search for and track a specified target (Fig. 1b).

A viable possibility is to separate the component that identifies the target in the images from that one responsible for planning the robot motion. However, the non-trivial problem of combining the two components remains. For this reason, more recent works [7], [8] propose the use of DRL algorithms to address the problem in an end-to-end manner.

Since the majority of these approaches consider only discrete actions policies, we propose a novel DRL-based architecture for VAT in *continuous* action spaces, which, even if trained with synthetic data only, can be effectively used in real scenarios with physical robots.

II. APPROACHES AND TASKS DETAILS

To train our models we build synthetic environments by using the photorealistic graphics engine Unreal Engine 4 (UE4)[1], which allows to design and customize complex 3D worlds. The environments for TDVN consist of 3D mazes where the agent and the goal are randomly placed, while the ones for VAT are large empty rooms, within which both the *tracker* and the *target* are positioned. Since our methods are both trained in simulated environments only, in order to

[1]https://www.unrealengine.com

achieve generalization also to real world contexts, we apply *domain randomization* [9] to our synthetic scenarios.

A. Target Driven Visual Navigation

The TDVN problem consists in finding the shortest sequence of control 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.

The observation consists of the current RGB frame from the agent point of view and the image of the target to be reached. Both inputs are 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. We use our dataset collected in simulation, whose samples consist of triplets of images, each containing: the picture of the goal, an image in which the goal is visible and an other one in which it is not. The navigation network is trained via DRL using IMPALA [10].

B. Visual Active Tracking

The objective of an autonomous robot that performs VAT, referred to in the following as *tracker*, is to recognize and actively track a predefined, and possibly moving, *target*, by using only visual inputs. In particular, in our setting, the *tracker* is free to move along the X and the Y axes of a three-dimensional space and to adapt its orientation to keep the *target* in the field of view of its camera (we assume it mounted in the front of the robot).

When an episode starts, the *tracker* has first to look around to find the *target*, since it can also spawn outside its initial field of view, then it can start to track it. The *tracker* action space is continuous. At each timestep t, it produces two different and independent real valued actions, a_{ρ_t} and a_{θ_t}, which represents the translational and the angular speeds, respectively.

III. EXPERIMENTS

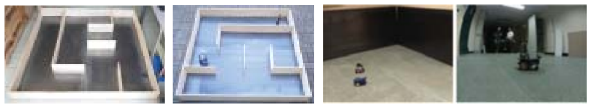
A. Target Driven Visual Navigation

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

1) Exploration Experiment: In this experiment, we place the agent in the center of a 20×20 maze, which is much larger than the 3×3 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.

2) Target-driven Experiment: In this second experiment, we place our agent in a 5×5 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.

To test our model performance in real settings (Fig. 2a, 2b), we build 7 different 4×4 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.



(a) (b) (c)

Fig. 2. Examples of (a) indoor and (b) outdoor mazes. In these two settings, the backgrounds are completely different, especially compared to those seen during training. (c) Two images captured from the recording camera (left) and the *tracker* robot point of view (right) during the real world experiments.

B. Visual Active Tracking

With the experiments, we want to asses our approach generalization capabilities and its robustness in real conditions. To this aim, we deploy our algorithm, without any kind of fine-tuning, in an indoor environment on a real robot (Fig. 2c). Both the *tracker* and the *target* robots can perform the same actions as their simulated counterparts. The former is controlled by our model residing in a remote host, while the latter is manually controlled.

Although our methods exhibits lower performance than those in simulation, it is still able to achieve remarkable results also in a real world environment. It should be noticed that this test scenario significantly differs from the simulated ones in terms of visual appearance. In particular, it is characterized by various objects, people, textures and lightning conditions that are completely absent in the training environments. Despite that, the robot is still able to recognize and locate the target.

REFERENCES

[1] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on systems, Man, and Cybernetics*, 1989.

[2] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, 2017.

[3] Y. Zhu *et al.*, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017.

[4] X. Ye *et al.*, "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.

[5] F. L. Lewis *et al.*, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Systems Magazine*, 2012.

[6] D. Zhang *et al.*, "Deep reinforcement learning for visual object tracking in videos," *arXiv preprint arXiv:1701.08936*, 2017.

[7] W. Luo *et al.*, "End-to-end active object tracking and its real-world deployment via reinforcement learning," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[8] F. Zhong *et al.*, "Ad-vat+: An asymmetric dueling mechanism for learning and understanding visual active tracking," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[9] J. Tobin *et al.*, "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.

[10] L. Espeholt *et al.*, "Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures," *arXiv preprint arXiv:1802.01561*, 2018.