Reinforcement Learning with Object Localization in a Virtual Environment

Joseph Gussev, Helge Ü. Dinkelbach, Frederik Beuth, and Fred H. Hamker

Technische Universität Chemnitz, Artificial Intelligence,
Strasse der Nationen 62, 09111 Chemnitz, Germany
jgus@hrz.tu-chemnitz.de

Abstract. Usually in reinforcement learning scenarios with complex environments, the agent is dependent on detailed external information, for example the location of objects to interact with. An idea would be to let the agent extract needed information from stimuli by itself. This study focuses on reinforcement learning in a virtual environment. It uses $Q(\lambda)$-learning with the main goal being to let an agent learn a goal-directed behavior. The agent learns to find relevant objects in its environment to be able to interact with them, using an attention-driven object localization module. Latter uses a recently developed model of the visual cortex and simulates the attentional processing of the primate brain. The agent in the implemented system is successfully able to use the ability of object localization in a virtual environment.

Keywords: reinforcement learning, virtual environment, object localization, visual attention

1 Introduction

Object recognition and reinforcement learning belong to the broad field of artificial intelligence, yet they are very different. While object recognition is a sub-domain of computer vision, reinforcement learning is a sub-domain of machine learning. Reinforcement learning methods are well discussed in different fields of application, for example robotics [1, 2], industrial manufacturing [3, 4] and computer game playing [5]. Many of these algorithms have a restricted sense or understanding of their environment, for example visual information about objects close to the agent. This work aims to provide the agent with the ability of object recognition, more specifically object localization, to close this gap. Instead of being dependent on detailed external information about the object, the agent would be able to extract needed information by itself from an image provided by a virtual environment. The goal of this study is to let an agent learn a goal-directed behavior, which is to interact with certain objects in a virtual environment. The system to fulfill this task consists of three essential components: reinforcement learning, object localization and a virtual environment. The attention-driven object localization system uses a recently developed model of the visual cortex and simulates the attentional processing of the primate brain.

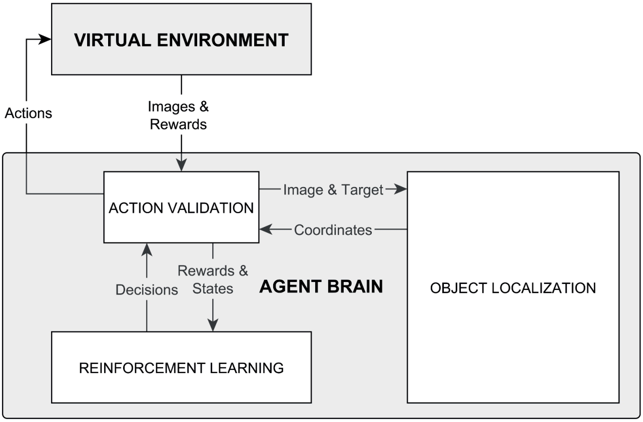


Fig. 1. An overview on the system structure and the information exchange between its components.

2 System Overview

The structure of this study's system as well as the information exchange between its components, which will be explained more detailed in this section, are illustrated in Fig. 1.

Reinforcement Learning is a class of iterative learning algorithms maximizing a long-term reward. It has been shown, for discrete state and action spaces, these algorithms can find optimal action policies for a given problem [6]. In most application scenarios, the reward occurs only after the task is completed, denoted by a terminal state. To deal with this delayed reinforcement, the eligibility trace mechanism [7] can be used. Every time an action is chosen, a trace is created for the corresponding action-state pair, which decays over time during the current trial. $Q(\lambda)$-learning [6] combines both Q-learning [8] and eligibility traces. Latter differs between two types: In accumulating traces [6], the trace gets accumulated each time the according action-state pair occurs. On the other hand, replacing traces [9] set the trace to an initial value each time instead of accumulating it. Whenever a trial ends or an exploratory action was chosen, all traces are deleted. One has to decide depending on the state and action space as well as the reward function which mechanism fits the scenario best. This study uses $Q(\lambda)$-learning with accumulating traces and a softmax action selection [6, 10].

Action Validation: To save up on simulation time as well as prevent the agent from performing absurd actions, for example reaching for an object while already grasping another one, an action validation was implemented. It takes the role of an interface between the agent and its environment. Based on the concept of an internal environment entity [11], this interface intercepts actions manually characterized as invalid and prevents them from being performed in the virtual environment. Instead, it sends a negative reward to the agent.

Object Localization: The task of object localization is to search a given target object in an image [12]. For that, a recently developed attention model [13] is used. The model is based on the primate visual system of the brain [14, 15], which consists of a bottom-up and top-down processing network. The relevant areas are the prefrontal cortex (PFC), the frontal eye fields (FEF) and the ventral steam, consisting of the primary visual cortex (V1), the fourth visual cortex (V4) and the inferior temporal cortex (IT). Each area encodes different information regarding the object as well as the task. The core of this model is the concept of visual attention, which is a selection process to select relevant information among a large amount of incoming sensory data [16]. Beuth and Hamker [13] propose that attention is a cognitive and holistic top-down control process, tuning the visual system for the current task. What attention does in this model is to amplify the response of neurons encoding task relevant stimuli and to suppress the response of neurons encoding the irrelevant stimuli, therefore adjusting neuronal activity for a current task.

Virtual Environment: To provide the visual stimuli needed for the object localization, the cross-platform game engine Unity was used as a virtual environment. For this study a Unity scene of the EU-project *Spatial Cognition* of the Professorship Artificial Intelligence was used, together with its agent 'Felice'. Using camera objects as eyes, Felice is able to receive visual information in the form of images from the environment. Information exchange between Unity and the rest of the system was done via network.

3 Task

Scenario, States & Actions: Spawning besides an open box the agent will face a table with three different objects on it (Fig. 2). The agent has to put any two objects in the box, one by one, to end a trial. The states and actions were modeled to represent elemental steps of this task. Hence, the agent is able to choose from the following action space:

1) MOVE to table
2) MOVE to box
3) GRAB center of view
4) LET go
5) VISUAL SEARCH (1st object)
6) VISUAL SEARCH (2nd object)
7) VISUAL SEARCH (3rd object)

A visual search lets the agent fixate its eyes on an object in the image received from the virtual environment. When standing away from the table, the action also sets it as the target. Although fixation is lost when the agent moves towards

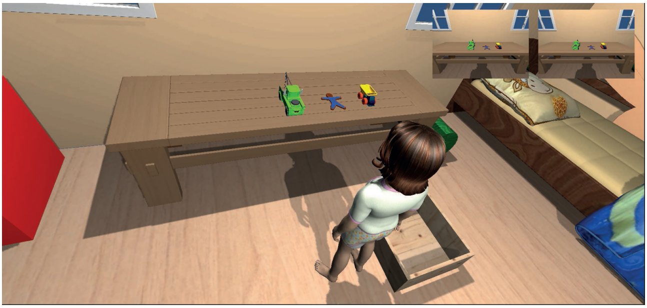


Fig. 2. Agent 'Felice' and the three objects chosen for the study. This constellation represents the initial state.

the table, it makes sure the target object is still within the visual field to be fixated again and grabbed. A state is represented by a set of variables:

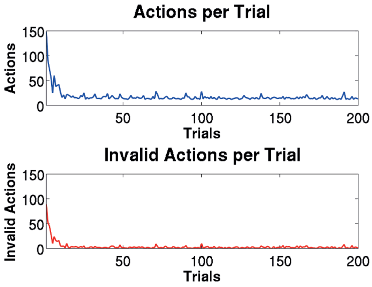
1) POSITION (1 = at the table; 2 = at the box)
2) HANDS (0 = emtpy; 1-3 = grabbing 1st / 2nd / 3rd object)
3) TARGET OBJECT (0 = emtpy; 1-3 = 1st / 2nd / 3rd object)
4) FIXATED (0 = not fixated; 1 = fixating target object)
5) BOX (0 = emtpy; 1-3 = 1st / 2nd / 3rd object; 4 = two objects)

Reward Function: Whenever the action validation intercepts an invalid action, a reward of $r = -11$ is sent to the agent. The rewards of actions performed within the virtual environment are received from latter itself, only yielding positive reward whenever objects are dropped into the box. Object 1 yields $r = 20$, Object 2 $r = 10$ and Object 3 $r = 30$. As soon as two objects are within the box, signalling the agent's terminal state, the agent receives a final reward consisting of the sum of the rewards of the corresponding objects. Every other action is rewarded with $r = -1$.

Parameters: The $Q(\lambda)$-learning parameters for this scenario were set with learning rate $\alpha = 0.1$, discount $\gamma = 0.7$ and trace decay $\lambda = 0.8$, the softmax parameters with temperature $T = 30$ and exploration rate $\epsilon = 0.2$.

4 Results

Learning converged after about 20 trials: Fig. 3 shows the number of actions as well as the number of invalid actions taken by the agent, per trial. Performing around 150 actions per trial at the beginning, with around 90 of them being invalid, the agent quickly sorts those invalid actions out. The reward of $r = -11$ of invalid actions compared to $r = -1$ of the other actions is responsible for the decrease of invalid actions, since their Q-Values decrease more rapidly with a



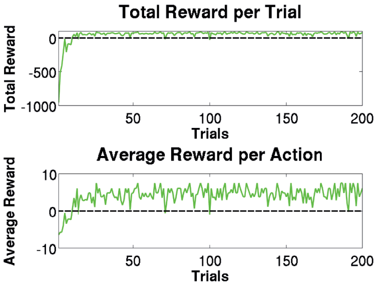


Fig. 3. Actions and invalid actions per trial after 200 trials.

Fig. 4. Total reward and average reward per action after 200 trials.

higher negative reward. After 20 trials, the number of actions per trial averages 25, with only about 0 to 10 being invalid. Latter are still existent because of two reasons: The first is simply exploration. The second one would be, that the states near to the initial state are visited often, so the according Q-Values can be frequently updated for the agent to effectively avoid invalid actions in those states. Meanwhile, the states near the terminal state are visited less, resulting in the Q-Values of the negative actions being not as established. The convergence of learning is also indicated by the total reward and average reward per action, plotted in Fig. 4. Starting with a reward of around -900 per trial and an average reward of -6 per action as a result of the majority of chosen actions being invalid, the agent adjusts to positive rewards in both total reward tending towards the maximum of 90 per trial and average reward per action being at about 6. Since the agent learned to avoid the invalid actions, the total reward per trial as well as average reward per action allow the positive rewards to stand out. This is a clear sign, that the agent learned to go for the profitable actions instead of the invalid and punishing ones. The behavior the agent learned and will use when choosing greedily after 200 trials is represented by the following action chain: 7, 1, 7, 3, 2, 4, 5, 1, 5, 3, 2 and 4. This results in the maximum total reward and average reward per action and shows, that the agent learned an optimal behavior.

5 Conclusion & Outlook

Concluding this work, the idea of a system combining reinforcement learning, a virtual environment and object localization was implemented. The Q-Values converged towards the intended behavior using $Q(\lambda)$-learning with accumulating traces and a softmax action selection mechanism. A possible idea for future research based on this work would be replacing object localization with object detection and accordingly creating a more complex state and action space with an active change of the agent's visual field being possible.

References

1. Takahashi, Y., Asada, M. and Hosoda, K.: Reasonable performance in less learning time by real robot based on incremental state space segmentation. In: Intelligent Robots and Systems '96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on, Volume 3, pp. 1518-1524. IEEE. (1996)
2. Takahashi, Y. and Asada, M.: Vision-guided behavior acquisition of a mobile robot by multi-layered reinforcement learning. In: Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on, Volume 1, pp. 395-402. IEEE. (2000)
3. Ueda, K., Hatono, I., Fujii, N. and Vaario, J.: Reinforcement Learning Approaches to Biological Manufacturing Systems. CIRP Annals Manufacturing Technology, Volume 49, pp. 343-346. (2000)
4. Mahadevan, S. and Theocharous, G.: Optimizing Production Manufacturing using Reinforcement Learning. In: Proc. 11th International FLAIRS Conference, pp. 372-377. AAAI Press. (1998)
5. Defazio, A. and Graepel, T.: A Comparison of learning algorithms on the Arcade Learning Environment. arXiv preprint arXiv:1410.8620 (2014)
6. Sutton, R. S. and Barto, A. G.: Reinforcement Learning: An Introduction. Cambridge, MA: MIT Press. (1998)
7. Klopf, A. H.: Brain function and adaptive systems: a heterostatic theory. Technical Report AFCRL-72-0164. Air Force Cambridge Research Laboratories. (1972)
8. Watkins, C. J. C. H.: Learning from delayed rewards. PhD Thesis, University of Cambridge, England (1989)
9. Singh, S. P. and Sutton, R. S.: Reinforcement Learning with Replacing Eligibility Traces. Machine Learning, Volume 22, pp. 123-158. Kluwer Academic Publishers-Plenum Publishers. (1996)
10. Tan, M.: Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents. In: Proc. 10th International Conference on Machine Learning, pp. 330-337. (1993)
11. Singh, S. P., Barto, A. G. and Chentanez, N.: Intrinsically motivated reinforcement learning. Saul, L. K., Weiss, H. and Bottou, L. (Eds.), Advances in Neural Information Processing Systems, Volume 17, pp. 1281-1288. MIT Press. (2005)
12. Wolfe, J. M.: Guided search 2.0 a revised model of visual search. Psychonomic Bulletin & Review, Volume 1, pp. 202-238. (1994)
13. Beuth, F. and Hamker, F. H.: Attention as cognitive, holistic control of the visual system. In: Villmann, T. and Schleif, F. M. (Eds.), Proc Workshop New Challenges in Neural Computation 2015 - NCNC 2015, Machine Learning Reports 03/2015,pp. 133-140. (2015)
14. Jamalian, A. and Hamker, F. H.: Biologically Inspired Models for Attentive Robot Vision: A Review. In: Pal, R. (Ed.), Innovative Research in Attention Modeling and Computer Vision Applications, pp. 69-98. Information Science Reference. (2016)
15. Serre, T.: Learning a dictionary of shape-components in visual cortex: Comparison with neurons, humans and machines. PhD Thesis. (2006)
16. Carrasco, M.: Visual attention: the past 25 years. Vision Research, Volume 51, pp. 1484-1525. (2011)