Interactive Segmentation using Object Singulation

Perception and Learning for Robotics

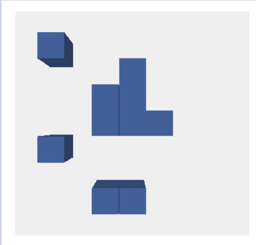
Patrick Pfreundschuh and Simon Schaefer

Supervised by: Fadri Furrer, Tonci Novkovic, Michel Breyer

Autonomous Systems Lab

Introduction

Manipulating unstructured environments is a major challenge in robotics, since often objects can only be viewed partially, which makes segmentation hard.



To simplify segmentation under these conditions, the objects can be singulated by pushing movements. To do so, the robot has to explore, if a segmented area really consists of a single object, or if it might be multiple objects side by side.

The infinite set of possible states and continuous actions as well as hardly predictable state transitions exacerbate deterministic modeling so that a reward-based learning approach is used. Previous approaches tackle object singulation by discretizing the state space, only allowing a limited number of actions. Within this project the feasibility of continuous approaches for object singulation is explored using Proximal Policy Optimization (PPO).

Method

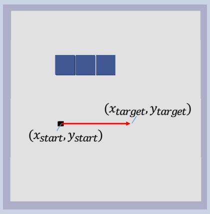
Native Approach

State: Binary map indicating object and background

Action: $a_{start\_target} = (x_{start}\ y_{start} x_{target}\ y_{target})$

Reward: $R = R_{DIFF\_DIS}$

➢ $R_{DIFF\_DIS} = h(\sum d_{i,j})$ rewarding an increaseing distance between objects (success for $\sum d_{i,j} > g(d_{max}, n_{objects})$)



Three-block scenario

Choosing a starting point close to an object most probably leads to a good pushing action. The action space mostly consists of actions not touching any object, ending up in very sample-inefficient training. Therefore, an advanced approach was impelemented addressing this problem:

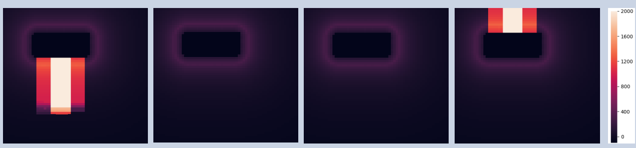
Advanced Approach

State: Euclidean distance map indicating the distance to the closest object based on the camera input image, offering gradient information about the closest object

Action: $a_{cardinal} = (x_{start}\ y_{start}\ c), c \in [\downarrow, \uparrow, \rightarrow, \leftarrow]$ simplifying the action space without large loss of generality in simulated ("cubic") world

Reward: $R = R_{START} + R_{DIFF\_DIS}$

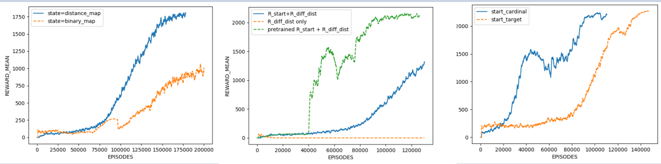
➢ $R_{START} = f(p_{start}, p_{closest-object})$ giving an incentive to start close to an object, penalty for starting inside an object



Reward heatmaps for all cardinal pushing directions for scenario above, mapped by starting point (from left to right: up, right, left, down)

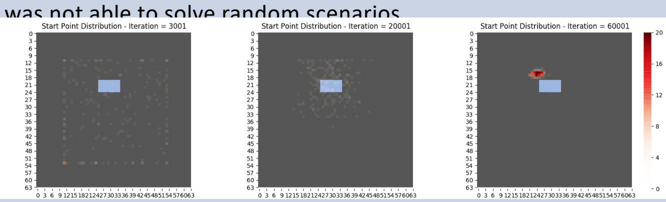
Experiments

In the following the impact of the advanced approach is compared to the native approach for a scenario where two connected blocks are always positioned at the same position. The reward curves show that our approach (blue) converges faster (or enables converging to a optimal solution at all) to an optimal solution.



Distance vs Binary Map   *R_{START} pretraining on R_{START}*   *$a_{cardinal}$ vs a_{start_target}*

In general object singulation tasks, the reward usually is multimodal. However PPO uses unimodal gaussian policies. In the task below, the agent was trained to spawn close to the object. As seen below PPO first tries to capture this multimodal distribution with a unimodal policy with high variance and mean inside the block, which complicates converging to one high-rewarded action. In a scenario where blocks are placed randomly and also including push directions, this behaviour becomes even more problematic, such that for our object singulation task PPO was not able to solve random scenarios.



Evolution of action start point distribution during training using PPO

To show that the task is solvable using RL we were able to solve a task in which two connected blocks are placed randomly using Deep Q Learning (DQL).

Conclusion & Future Work

Although some improvements in sample-efficiency of a continuous DRL approach using PPO for the object singulation task were shown, it could not be applied to complex tasks requiring a multimodal policy distribution. However, simple tasks could be solved using a DQL based approach in simulation. Possible further improvement and future applications are:

- Learn a more dense representation of the observation using an autoencoder design
- Use (more) real-world scenarios, e.g. robot actuator or real-world objects
- Apply as presegmentation step in object manipulation tasks

References

- A. Eitel, N. Hauff, and W. Burgard. Learning to Singulate Objects using a Push Proposal Network. 2017.
- W. R. F. Zeng, Song. Learning synergies between pushing and grasping with self-supervised deep reinforcement learning. 2018.
- Y. Tang, S. Agrawal. Implicit Policy for Reinforcement Learning. 2019