Object Recognition and Computer Vision
Project proposal on Topic F

Matias Etcheverry
Ecole Normale Supérieure
matias.etcheverry9@gmail.com

1. Topic & Working environment

Pick-and-place is a common task performed by robots in manufacturing. The goal is to pick an object, possibly rotated and put it in a different place. Classic systems use pose estimation of known objects as well as scripted planning. These systems are known to require copious amounts of data and suppose object-centric assumptions by computing key points or dense descriptors. The work proposed by Zeng et al. [3] proposes a new method to deal with multiple robotics tasks, reaching state of the art results, while maintaining an excellent sample efficiency.

Google-research proposed a new environment called `Ravens` [2] which is a collection of simulated tasks, coded like `Gym`. It allows to compare algorithm in a fixed environment. 10 types of simulated tasks are available, like packing boxes or sweeping piles of small objects. The simulated tasks are formulated as Markovian, where the current state of the tasks only depends on its previous state.

2. Datasets

I will be focused on the block insertion and the manipulating rope tasks. The first one aims at picking up an L-shaped red block and placing it into an L-shaped fixture with a corresponding hole, while the objective of the second task is to manipulate a rope so that it finishes the incomplete perimeter of a square. The authors of the article have provided precomputed datasets of the 2 simulated tasks. Using these datasets ensures that I will be working with the same data as the article. Each dataset is made of 1000 training and 100 test demonstrations. Each demonstration gathers the beginning, intermediate and the final state of the task, where each state is composed of an RGB-Depth image and a reward. The observation also depicts the true actions performed by the robot.

3. Plan of work

The goal of this project is to explore the capabilities of Transporter Networks. Thus, my work on this project will be twofold. Firstly, I will train multiple Transporter Networks to see if they perform as good as in the article. Then, I will try to get rid of the depth input data.

3.1. Metrics

In order to check if my implementation and the experimentations are meaningful, I will need to use specific metrics to compare models. Firstly, I will use the task success rate on the whole test set. It is also interesting compute the number of actions taken to complete a task. Moreover, we can also measure how good the pose of a displaced rope is. To do so, we determine the overlapping area of the displaced rope with its target pose. In the end, it is also necessary to see if those policies which are meant to be performed by real robots are qualitatively right: no loops nor strange movements for instance.

3.2. Implementation

The first objective is to be able to generate policies on the two above mentioned tasks to reproduce the results reported by Zeng et al [3]. The Transporter Networks seem to be sample efficient. I will check this by training 3 different Transporter Networks using respectively 1, 10 and 100 observations. Exploring this new environment, running it on virtual machine and training different models on different tasks should take me 12 hours (without taking into account the training time).

3.3. Experiments

Transporter Networks require RGB-Depth images. Other solutions like *GT-state MLP* even use 3D bounding boxes to generate policies. While obtaining depth images is getting easier, it is still preferable to only use RGB images. Thus, I will check how the Transporter Networks behave when using only RGB images. First of all, I will delete the depth information, by providing the same constant depth to the network. The network won't be able to infer any information from depth. Then, I will apply monocular depth estimation using the BinsFormer framework [1]. This will allow me to estimate the depth on every camera images used by the robot. Comparing models with such conditions will likely take me 10 to 17 hours.

References

[1] Zhenyu Li, Xuyang Wang, Xianming Liu, and Junjun Jiang. Binsformer: Revisiting adaptive bins for monocular depth estimation, 2022. 1

[2] Google Research. Ravens - transporter networks. https://github.com/google-research/ravens, 2020. 1

[3] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Ayzaan Wahid, Vikas Sindhwani, and Johnny Lee. Transporter networks: Rearranging the visual world for robotic manipulation, 2020. 1