Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection

Reminders

- “Hand-eye coordination is the coordinated control of eye movement with hand movement, and the processing of visual input to guide reaching and grasping along with the use of propriception of the hands to guide the eyes.” - Wikipedia
  - Here the camera is fixed though
- Visual servoing - using vision to provide closed-loop position control for a robot end-effector
  - The alternative is open-loop “looking” then “moving” (more common of course)

Overview

- The goal is to learn actions that resut in good grasps and to achieve the learning algorithm was applied on 14 robots
- First a classifier is used to learn good grasps which are then used to apply an optimization algorithm to infer best actions

Key ingredients

- Network $g(I_t, v_t)$ is trained to predict whether a given task-space motion v_t would result in a successful grasp, based on the current camera observation I_t
- Then cross-entropy method is performed to optimize over v_t (even though just sampling and choosing the best one gives results)
- What is interesting are the training samples. The gripper is moved around every time step, so v_t is changing, but only at time T the grasp is evaluated.
  - The training data contains $(I_t, p_T - p_t, \text{success})$, so it keeps the difference between final and current position, and not the next and current which is v_t.
  - In case of a good grasp at p_T, we don’t get anything from knowing where we move next at p_t, only where we end up, so the best action is in the direction of where we end up - assumption.

Comments

- It is mentioned that the algorithm starts with random actions with $T = 2$. This means that the end effector moves for one time step and in the next step the gripper is closed and the grasp is evaluated.
  - I don’t see how this random exploration could yield good grasps - there must be some bootstrapping involved, to at least have “something working”
- The end effector is only kept in the vertical position, and it is argued that it is straightforward to generalize it. I wonder how many more samples would be needed to make learning efficient. Also, there are some heuristics that would not work out of the box there - like escaping the clutter by simply moving up...
- I wonder if the same procedure would be done in simulation, how meaningful would the results be? Is it worthwhile to employ so many robots for such long period of time?
- Does all the exploration/exploitation