Robot Learning
Weekly Exercise 9

Marc Toussaint & Wolfgang Hönig
Learning & Intelligent Systems Lab, Intelligent Multi-Robot Coordination Lab, TU Berlin
Marchstr. 23, 10587 Berlin, Germany

Summer 2024

1 Literature: Grasp Data Collection

Here is a core paper on grasp data collection:

H.-S. Fang, C. Wang, M. Gou, and C. Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11444–11453, 2020. URL: http://openaccess.thecvf.com/content_CVPR_2020/html/Fang_GraspNet-1Billion_A_Large-Scale_Benchmark_for_General_Object_Grasping_CVPR_2020_paper.html

The collection of labelled grasp data is a central issue in learning-based grasing. Once such data is available, we can use strong supervised ML or diffusion methods to learn disciminative or generative models of grasps. The above paper is a good example on how grasp data generation is typically "engineered", and uses a model-based (force closure) method to provide grasp labels. (An alternative is to use a generic physical simulator, e.g., [1] is a recent paper generating a grasp dataset using the PhysX simulator.)

The questions are only about Section 3.2 and 3.3:

a) Sec. 3.2 describes how 97,280 RGB-D images were taken. How is the camera pose known for each image? What are ArUco markers? For how many scenes were images collected?

Camera pose calibration is interesting: Cameras are mounted on a robot – while they trust reproducibility (same robot joints, same camera pose), they do not trust forward kinematics! So they calibrate the camera in each of the 256 poses using (a single?!) fiducial.

ArUco markers are partidular fiducial markers... https://en.wikipedia.org/wiki/Fiducial_marker

190 scenes; 2 cameras; 256 poses = 97280 images

b) Concerning Sec. 3.3 (paragraph "6D Pose Annotation"), how exactly are all 6D object poses annotated?

I think they don't say! Manually refined. Since they mention [17] several times; perhaps they use this as initialization.

c) Paragraph "Grasp Pose Annotation" is the core. Provide pseudo code to what is happening in the 2nd paragraph; make the looping over objects/points/anything explicit. (Section 5.2, 2nd paragraph provides the ranges of D, A, and V.) The last paragraph describes how these object grasps are transferred to the scenes. Summarize what information the eventual dataset comprises for one scene.

```
for o in objects:
  for p in points(o):
    for i=1..V:
      v ~ uniform on S^2
      for d in depths D:
        for a in angles A:
          w = smart choice (no collision)
          for mu = 0:0.1:1
            compute binary force closure
          s = 1.1 - smallest mu with closure
```

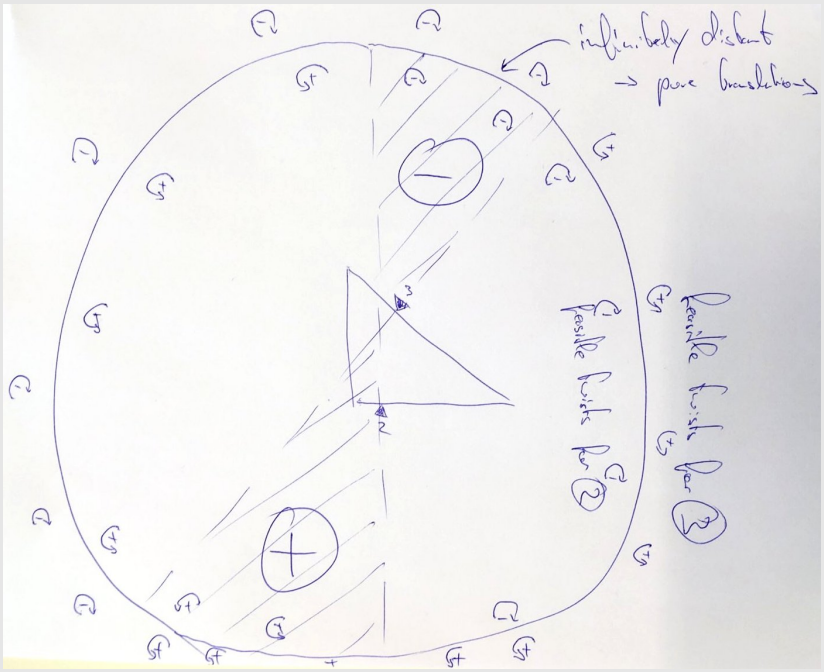
2 Force Closure

This is a great robotics book:

https://hades.mech.northwestern.edu/images/2/25/MR-v2.pdf

The Section "Grasping and Manipulation – Exercises" contains interesting force and form closure questions, around Fig. 12.29 and 12.30.

a) Solve Ex. 12.8 (page 507 in the pdf). Note that a twist in 3D space is a 6-vector combining a translation and rotation vector; here in 2D it is a 3-vector with 2D translation and one rotation. Sec. 12.1.6 (page 475) explains how to draw a twist as "CoR" – see footnote[1]



b) Solve Ex. 12.17. (I'll provide explicit equations defining force closure in the lecture.) (Ex. 12.18 is also a great exercise.)

The book explains that we have force closure iff both fiction cones can see the other point of contact. (That's much easier to evaluate than literally drawing the wrench cone and checking the origin is included.)

(a) The left cone has an 90-degree opening to the right; the right cone is actually the union of the 90-degree down cone and the 90-degree left cone - together they span 180 degrees. Both clearly see each other.

(b) The right cone changes: the *two contacts* of finger 2 are to the left and down. Both have no friction – so their cone is just a single line. But the span of both lines (left, and down) is a 90-degree cone to the bottom-left. Finger 1 is still just inside that cone, so we have force closure.

(c) Same situation: As long as finger 1 is inside the cone of finger 2, all is good. That is for positions $x = [0, L]$, then it starts slipping.

3 Practical Exercise: Explore the Graspnet data

This exercise doesn't need much coding – the aim is simply to familiarize youself with existing datasets and conventions for learning-based grasping.

a) Follow https://graspnetapi.readthedocs.io/en/latest/install.html to download and unzip all the data (sorry – lots of files... If you develop a script to do all downloads, share it with all students.)

b) Follow https://graspnetapi.readthedocs.io/en/latest/example_vis.html to visualize the grasp data. Automatically loop through all available objects (calling `showObjGrasp`), and all available scenes (calling `showSceneGrasp`).

What is the difference between format='rect' versus '6d'? (And why may it take minutes for format='6d'?)

[1] A convenient way to represent a planar twist $V = (v_x, v_y, \omega)$ (with rotation velocity ω, and translational velocities v_x, v_y) is as a **center of rotation (CoR)** at $(-v_y/\omega, v_x/\omega)$. An additional marker '+' or '-' tells if we rotate positively or negatively around this center.

c) The '6D grasp' documentation https://graspnetapi.readthedocs.io/en/latest/grasp_format.html#d-grasp explains how the grasp pose (translation and orientation) is stored. For a given scene (e.g. id=0), write a loop to output the grasp-translation and grasp-rotation-matrix for all grasps.

(What I do not understand: The Rectangle Grasp description seems to only describe grasps in the image plane – how it the real 3D rotation represented? Or it is not?)

References

[1] C. Eppner, A. Mousavian, and D. Fox. Acronym: A large-scale grasp dataset based on simulation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6222–6227, 2021. URL: https://ieeexplore.ieee.org/abstract/document/9560844/.

[2] H.-S. Fang, C. Wang, M. Gou, and C. Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11444–11453, 2020. URL: http://openaccess.thecvf.com/content_CVPR_2020/html/Fang_GraspNet-1Billion_A_Large-Scale_Benchmark_for_General_Object_Grasping_CVPR_2020_paper.html.