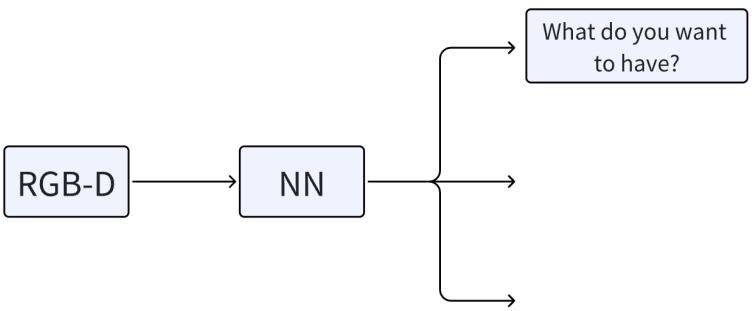
17. Deep Perception part2

Last Lecture: Traditional Computer Vision

- End to end, from pixel to joint torques

Today's Topic outline

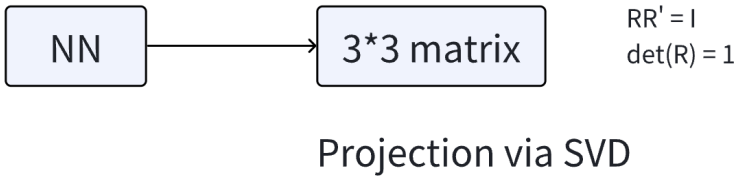
- What should we output for manipulation?
- Trends:
  - Transfer learning (start from pre-trained image net, via fine-tuning, to retarget network into our domain)
  - Synthetic data works pretty well
  - Self-supervised learning -> foundation models
- Possible intermediate representations
  - Pose (Deep pose Estimation)
  - Grasp score (for grasp selection)
  - Keypoints
  - Dense coorespondences
  - Many more



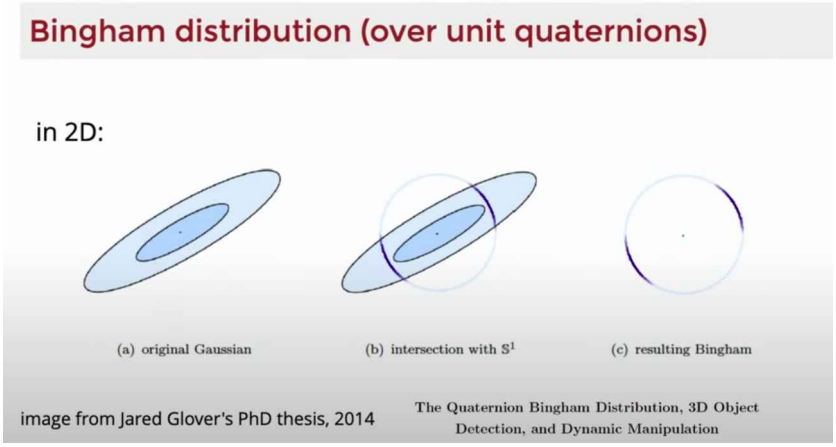
1. Deep pose estimation, for a known object

- How do you represent pose? (esp 3D rotations)
  - Discretize into bins -> classification (looks unreasonalbe at first, but good choice for PDF)

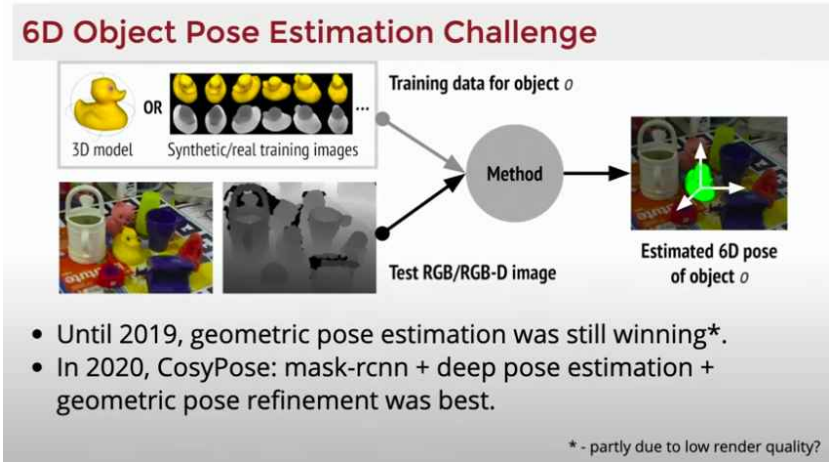
- R-P-Y (Eular angles) <- bad idea
- Quaternions
- Rotation Matrices
- How to choose Loss function?
  - Dont use rpy MSE distance, use quaternion (ger) distance
  - Shortest path for the rotation, shortest arc



- What about symmetries + Partial view?
- What we really want: a probability distribution of possible pose
- Bingham distribution (over unit quaternions), gaussian in the 4-D space quaternions
- Intersection of Gaussian and unit sphere



- Categorical distribution: https://bop.felk.cvut.cz/home/



- Limitation: for KNOWN objects

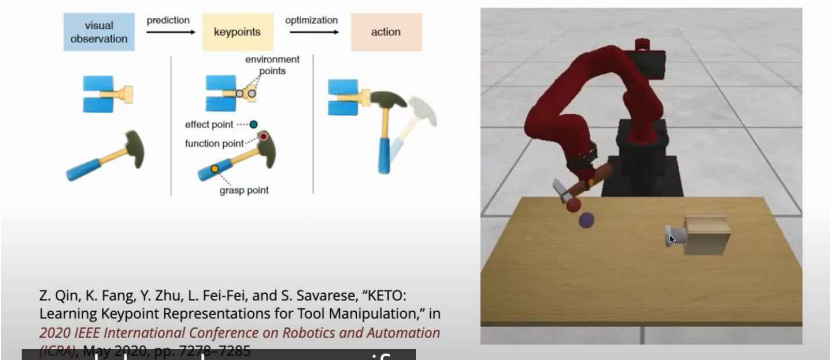
- Pose is so hard to estimate, and most of the time, is more than what you need to get the job done;
- Keypoints for boxes work good
- What about force control?
- Feedback Control for Category-Level Robotic Manipulation
- https://www.youtube.com/watch?v=GbblE4BtH08

2. Keypoints Affordances for Category Level Manipulation

- Keypoints are not sufficient representation
- Keypoint "semantics" + dense 3D geometry
- Human pose estimation 2d guide
- keypointnet.github.io/

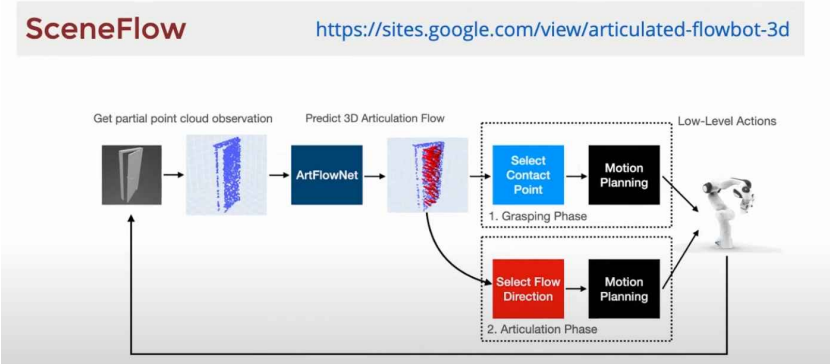
3. Force Control & Impedance Control?

- Feedback Control for Category-Level Robotic Manipulation:
- Need to know the points(Frames) that define impedances
  - Estimate the key points associated with orientation
  - Define the impedance, do peg insertion or wipping
- So far, the key points are geometric and semantic, required human labels.
- If we forgo semantics, can we self-supervise?



- Dense Object Nets, Core tech: dense correspondences
  - Dense Object Nets, Learning Dense Visual Object Descriptors by and for robot manipulation
  - Pick a point, scan, do the dense correspondances
  - SceneFlow
    - SceneFlow

- Take multiple images and do highlevel planning, get state of the environment object, where might the object move?
- Train a network to predict possible changes of the door



All examples today have one constraint. They need to be human interpretable, so that it can be connected to the manipulation pipeline