RoboTAP Implementation for PiH

00

Denoising Point Cloud Data

Denoising Point Cloud Data

Ideas:

- SuperQuadric Fitting
- Edge detection
- Unsupervised classification of stray points

Issues:

- Diffusion EDF may have model-inherent limitations — won't be solved even with perfect point cloud data
- Too much point-cloud manipulation can cause offsets in pose estimation.

01

Playing Around w/TAPIR

Summary of TAPIR

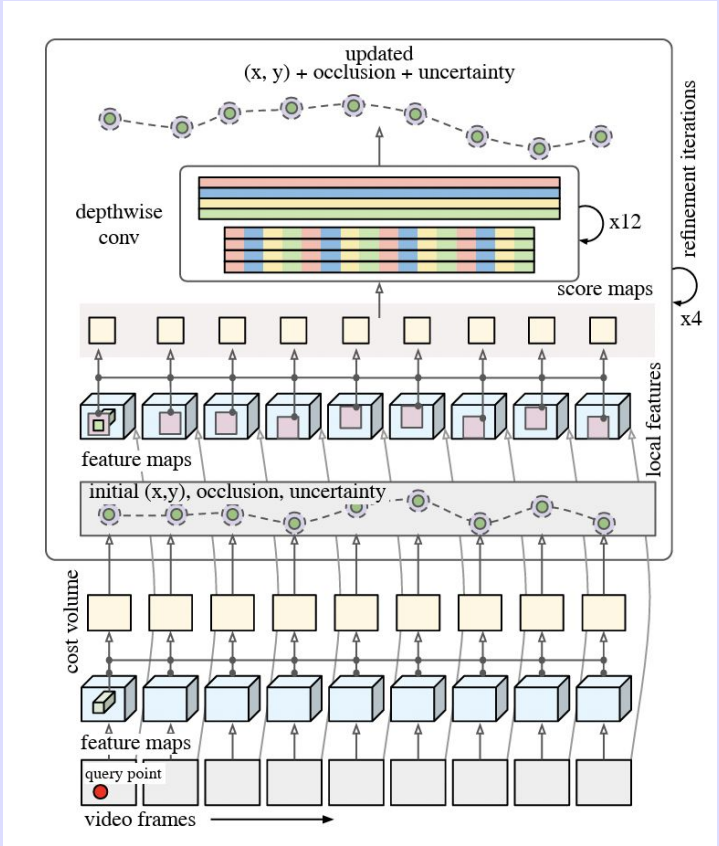
- Coarse-to-fine method ensures efficiency
- Iterative refinement ensures accuracy

Track Initialization:

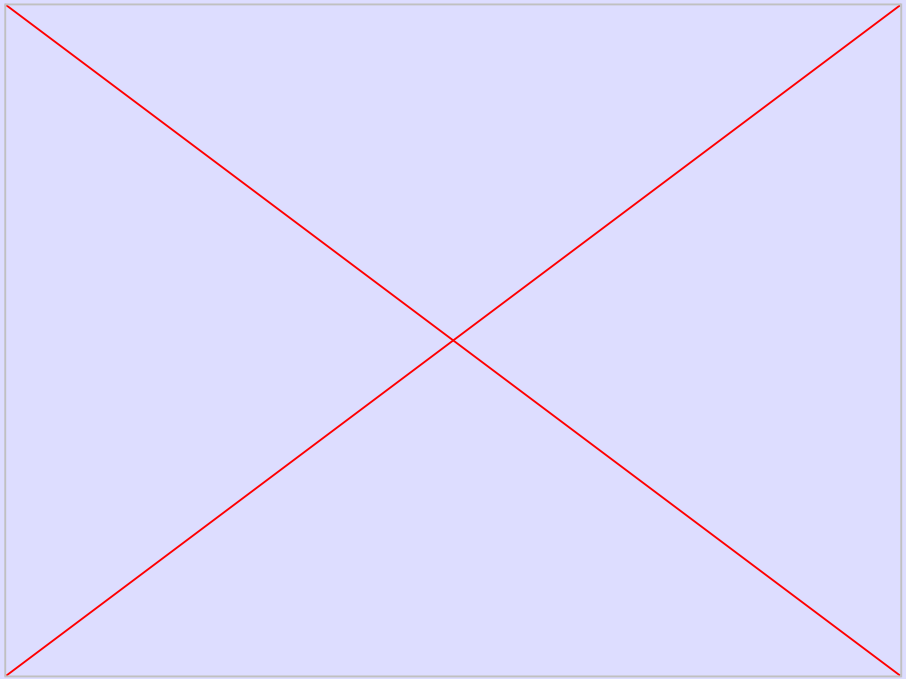
- Focuses on “global” features
- Features extracted via bilinear interpolation
- “Cost Volume” obtained from dot products of F_q with features on course map
- ConvNet produces heat map from cost volume for initial estimates of query positions and occlusion probabilities!

Iterative Refinement

- Looks at more local features (7x7 range) along track to update position and occlusion estimates.
- Integrates information across time (NOT causal).

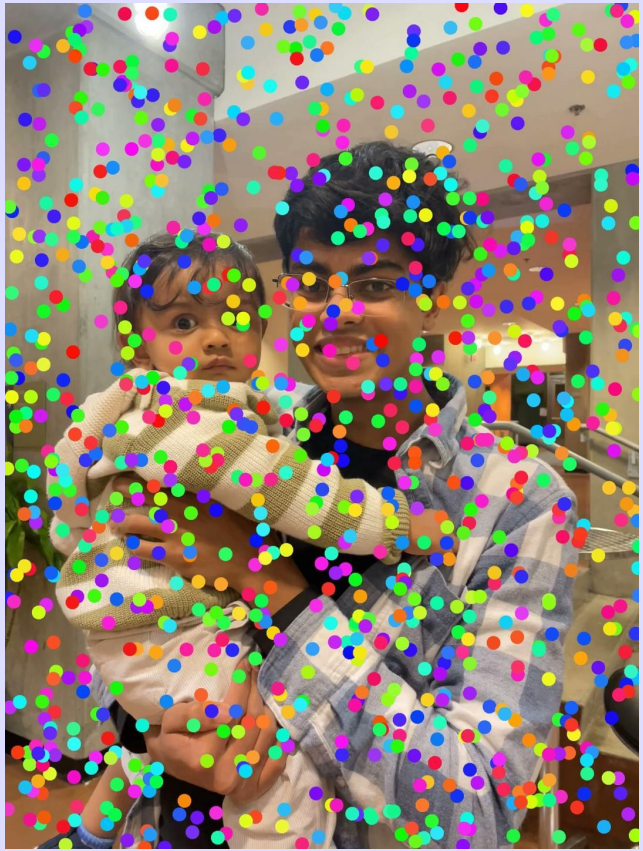


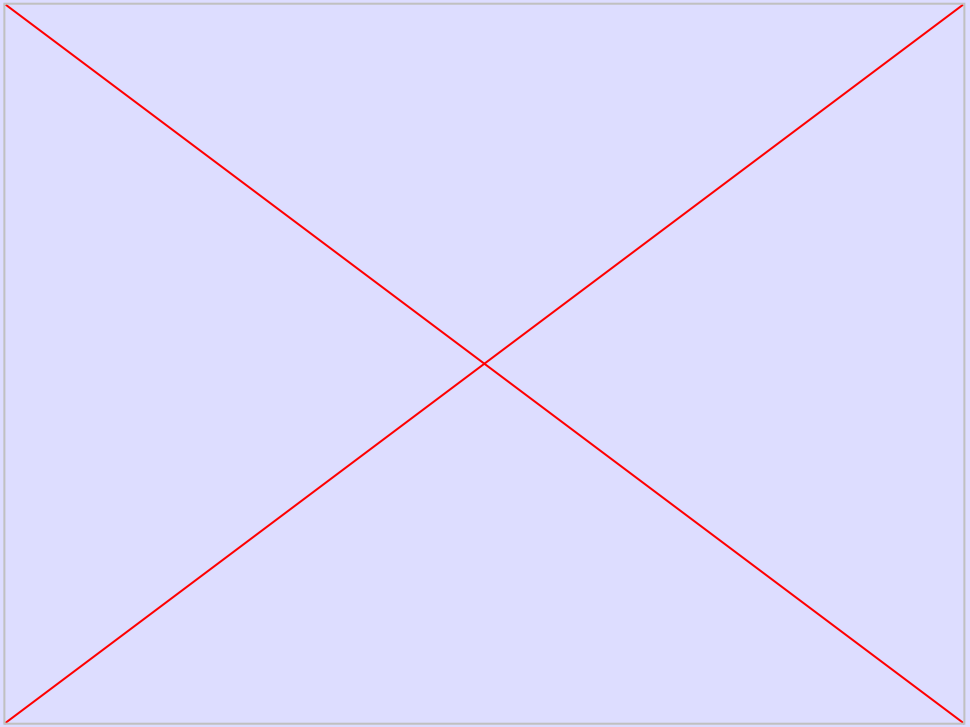
TAPIR Keypoint Tracking Demos



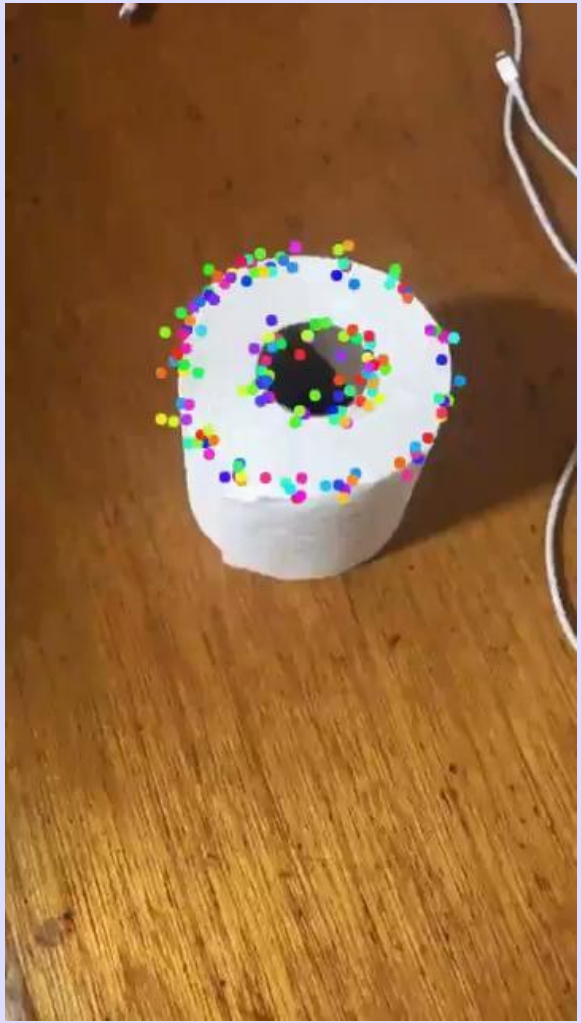
TAPIR Keypoint Tracking Demos (Horse)

TAPIR Keypoint Tracking Demos (Atharva)

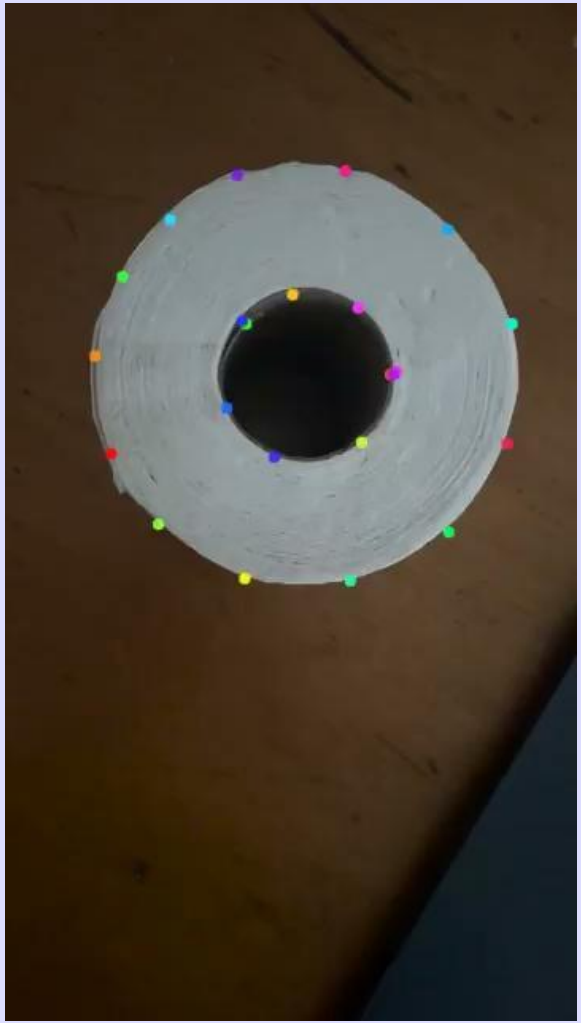




Robustness to Lighting



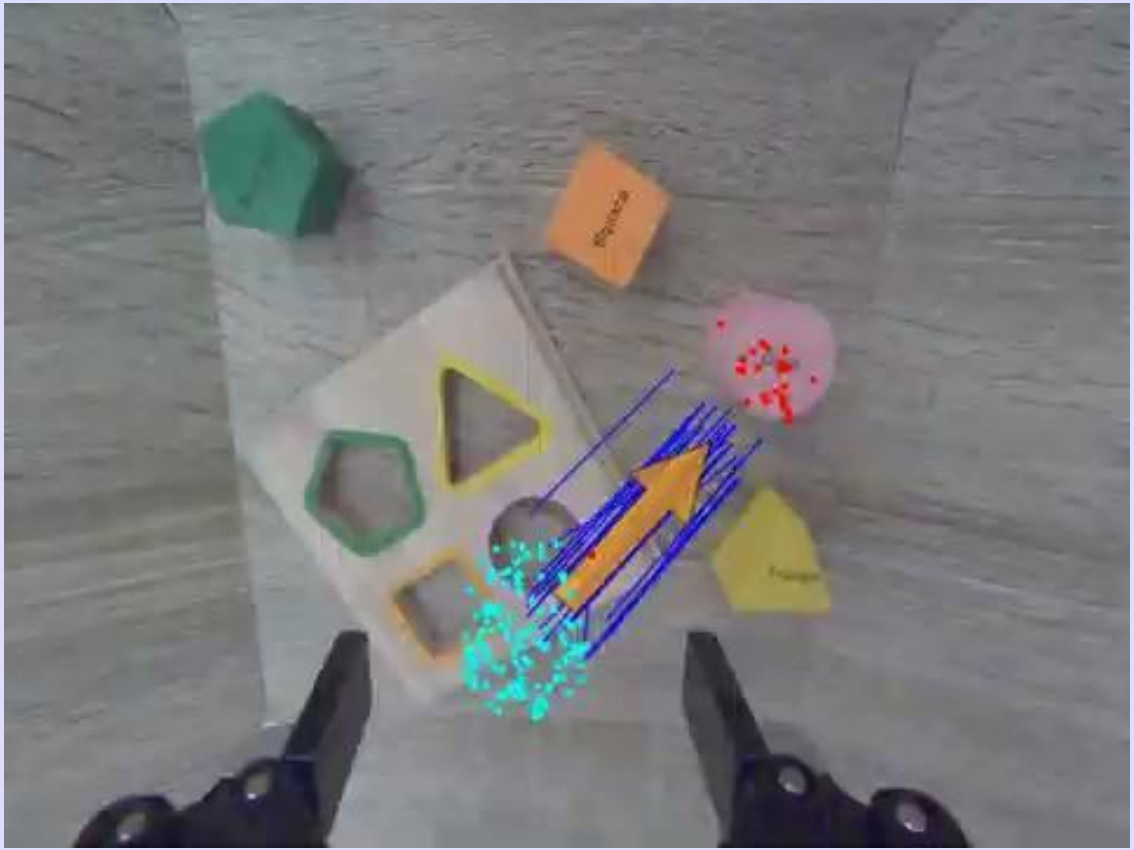
Robustness to Pose/Shape Change (3D)



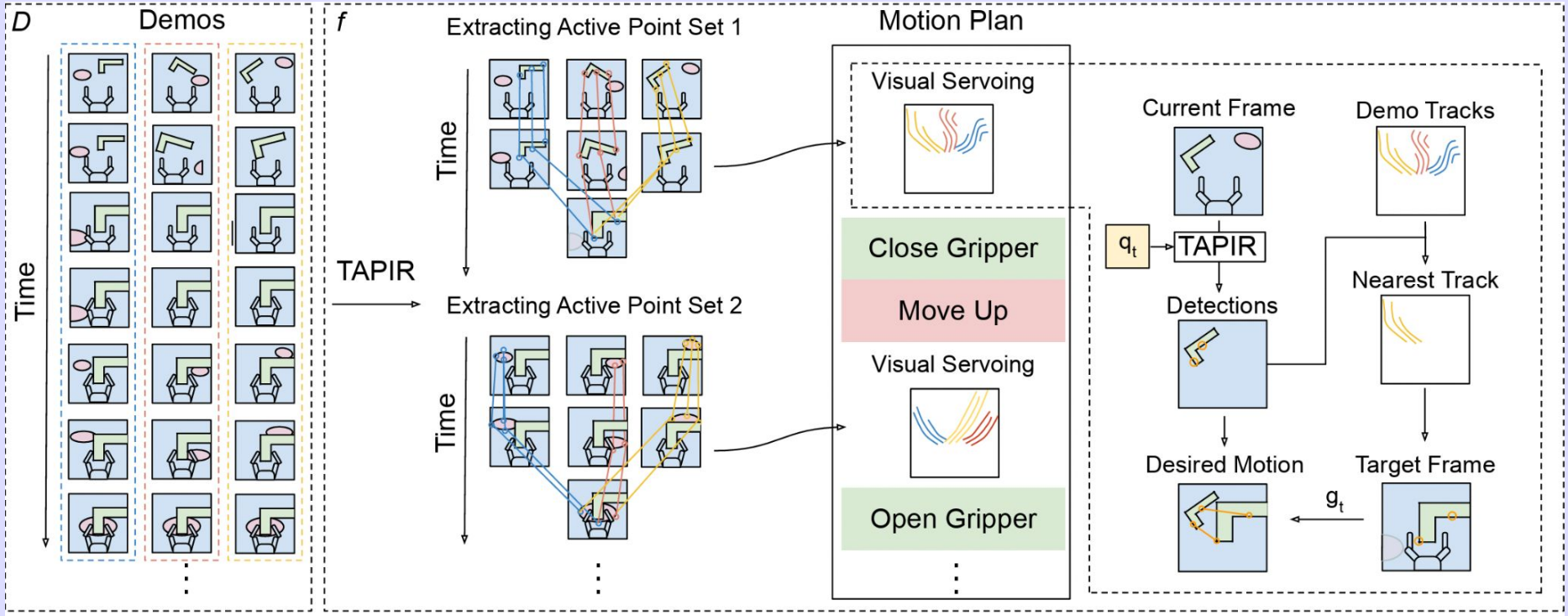
02

ROBOTap

RoboTAP



Full Pipeline



Overview of ROBOTap

$\pi(s_t, D) = \pi(TAP(s_t, q_t), g_t) = \pi(p_t, o_t, g_t)$ ——> Control policy is π

$g_t, q_t = f(s_t, D)$ ——> f is an active point selection algorithm (novel component)

$TAP(s_t, q_t) = p_t, o_t$ —-> TAP is TAPIR

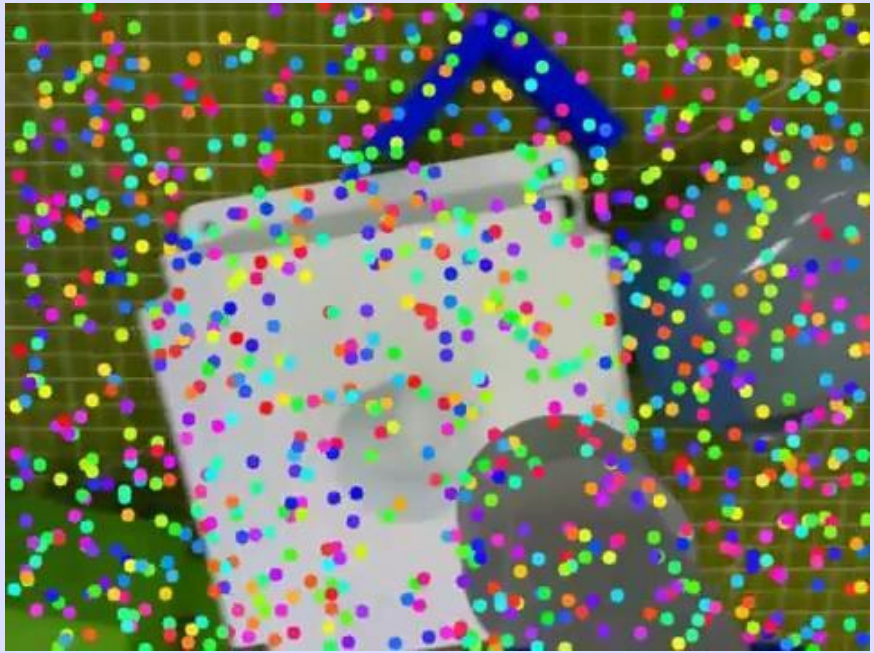
s_t: state image
D: demonstrations
q_t: queries obtained from demos (point features)

p_t: current positions of relevant points
g_t: target positions of relevant points
o_t: point occlusion probabilities

Clustering: $f(s_t, D)$

Combine the overlap between the low cross-demo variance (taken in final frame) and non stationary points to choose the most relevant motion clusters!

Trying it on Our Setup!



TAPIR

Clustering

Brief Reflection on Clustering: Why didn't it work well on our setup?

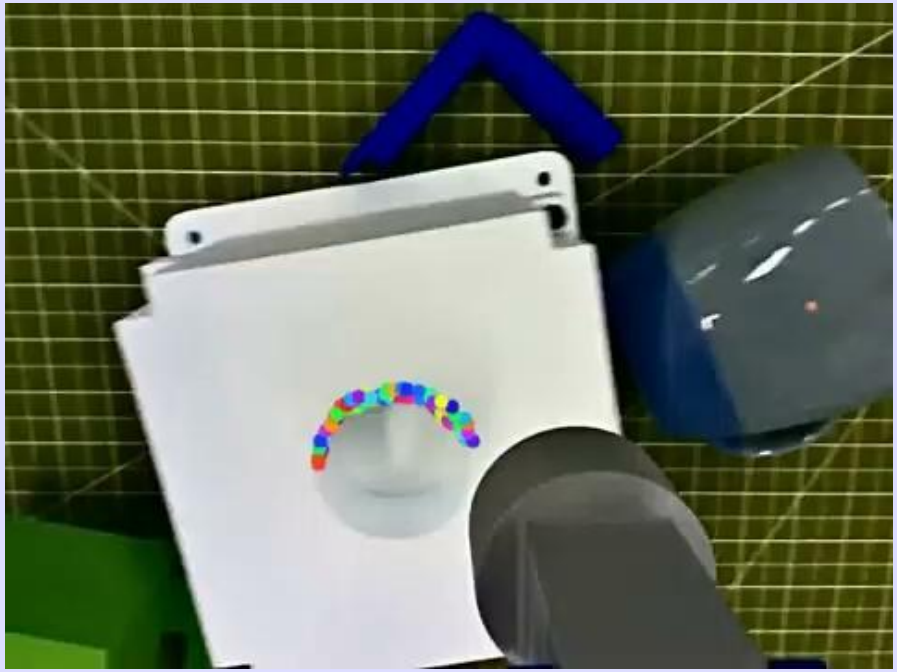
- Hole platform is featureless, preventing high quality detections.
- Detection of surrounding objects is mediocre, but unreliable (they won't necessarily be present).

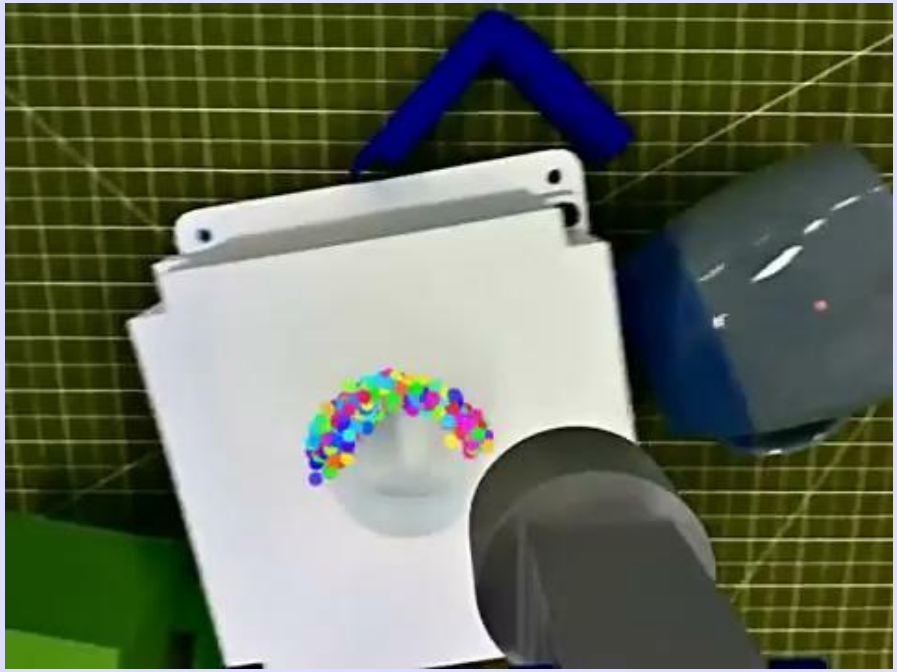
Solutions:

- Rather than using ROBOTap's clustering algorithm to select points, manually select points in the most task-relevant noising and randomly select surrounding points.
- Sharpen the image to add features!

Manually Selecting Query Points

Sharpening Filter, Im + Laplace(Im)





Overview

Pros

- Few demonstrations necessary.
- No retraining needed to obtain keypoints, minimal per-task engineering
- Closed-Loop
- Highly interpretable
- Robust to clutter and target pose randomization

Cons

- Lacks robustness to gripper pose
- Struggles with visual context-limited tasks
    - Fully occluded task-relevant object
    - Symmetries in goal

(Updated) ROBOTap Weaknesses to Consider

- Gripper pose
  - The peg may be in an orientation at which it is unable to be picked straight-on.
    - Manually selecting points for ROBOTap in the picking portion (and in general) would be problematic for cases in which selected queries aren't guaranteed to be visible!

1. We need to use ROBOTap in every temporal step in order to ensure consistency in starting frame
2. We have to either use clustering to select queries, or develop a pipeline that guarantees the visibility of manually selected queries.

- Struggles with visual context-limited tasks
  - Fully occluded task-relevant object (eg peg covers the hole)
  - Featureless goal
  - Symmetries in goal prevent the use of multiple demonstrations

1. Develop demonstrations that do not occlude points (adjustments to camera or demo trajectory).
2. Add our own features! Eg. scribble on the surface of the hole.

Next Steps (2/24)

- Identifying features and extracting descriptors
  - SIFT / Multi-Scale Oriented Patches
  - How to “average” descriptors, q_t, across multiple demos?
- Find target, g_t
- Active TAPIR to maintain updated features set
  - Temporal convnet -> causal convnet
- Control
  - Choose points with high certainty per TAPIR
  - Compute camera-frame transform
  - Convert to end-effector frame with image Jacobian

THANK YOU