Anthony Tran, Reya Dawlah,
Jason Ta, Jonathan Le

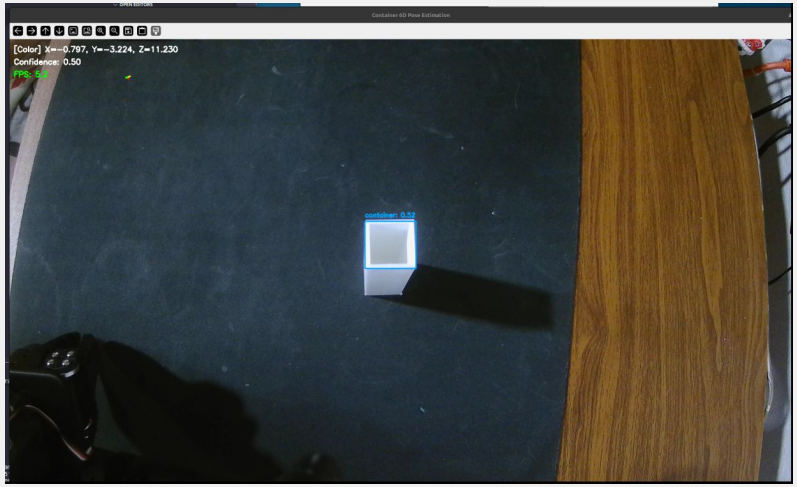
grab-n-pour

Problem overview

The robot must:

- Detect initial container (non-transparent)
- Estimate its 3D pose using the Perspective-n-Point (PnP) Algorithm
- Grasp the bottle
- Perform a spill-free pouring motion into a cup

Initial Methodology



Perception

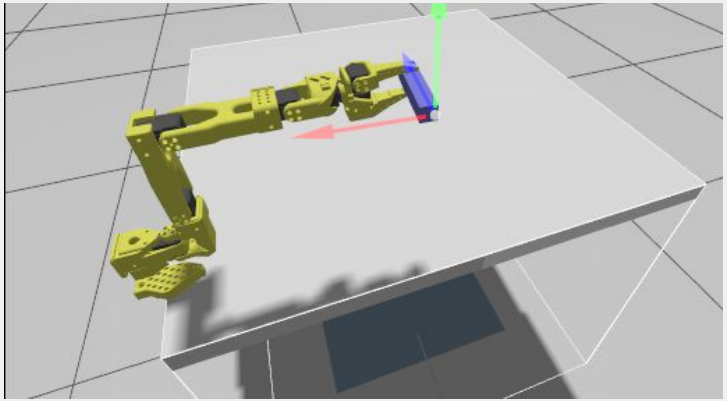
- RGB camera + YOLOv11
  - Detect cups /bottles
  - Confidence scores and bounding boxes
- Perspective-n-Point (PnP)
  - Converts 2D image keypoints to 3D coordinates

Motion Planning

- Moveit2
  - OMPL Planner
- RRTConnect
  - Pick_IK for inverse kinematics

Step 1: Simulation

- Used Gazebo and imported the SO-101 Arm
  - Lots of configuration issues -> Could not use motion planning in Rviz reliably
  - Using Trac_ik failed
  - Switched from KDL to pick_ik
  - Imported container STL and setup environment
- Used Moveit compute_ik to move robot to target position
- Once the simulation was working, we knew we had the correct transformation and we could focus on setting up TF correcting in the real robot environment.



Step 2: Physical Robot Arm

- Use of YOLO Model to detect container didn't work as expected
  - ➢ Due to a combination of the narrow design and bright white color
- Use of color-based model to detect container worked better
  - ➢ The contrast of the white container on black background helps the model with detection
- Perspective-n-point algorithm was inconsistent
- Took the bounding box location and used camera position to estimate the position of container via pixels
- Sent this container location via ros topic: /container_loc
- Python script takes this location data from /container_loc and transforms the coordinates from camera to robot frame
- Used Moveit to move to various configurations needed

Limitations

01 **Vision model heavily depends on white cups with a black background**

This is due to the current usage of the color-based model for container detection.

02 **Camera position must be same**

Since we assume the camera position and use the pixels for pose estimation.

03 **Simple grasp control that leads to inconsistent grasp variations**

04 **Simple pouring control that leads to inconsistent pouring and spillage**

Future Works

01

Improve vision detection

Using a depth camera or multiple cameras for better conversion of 2D to 3D

02

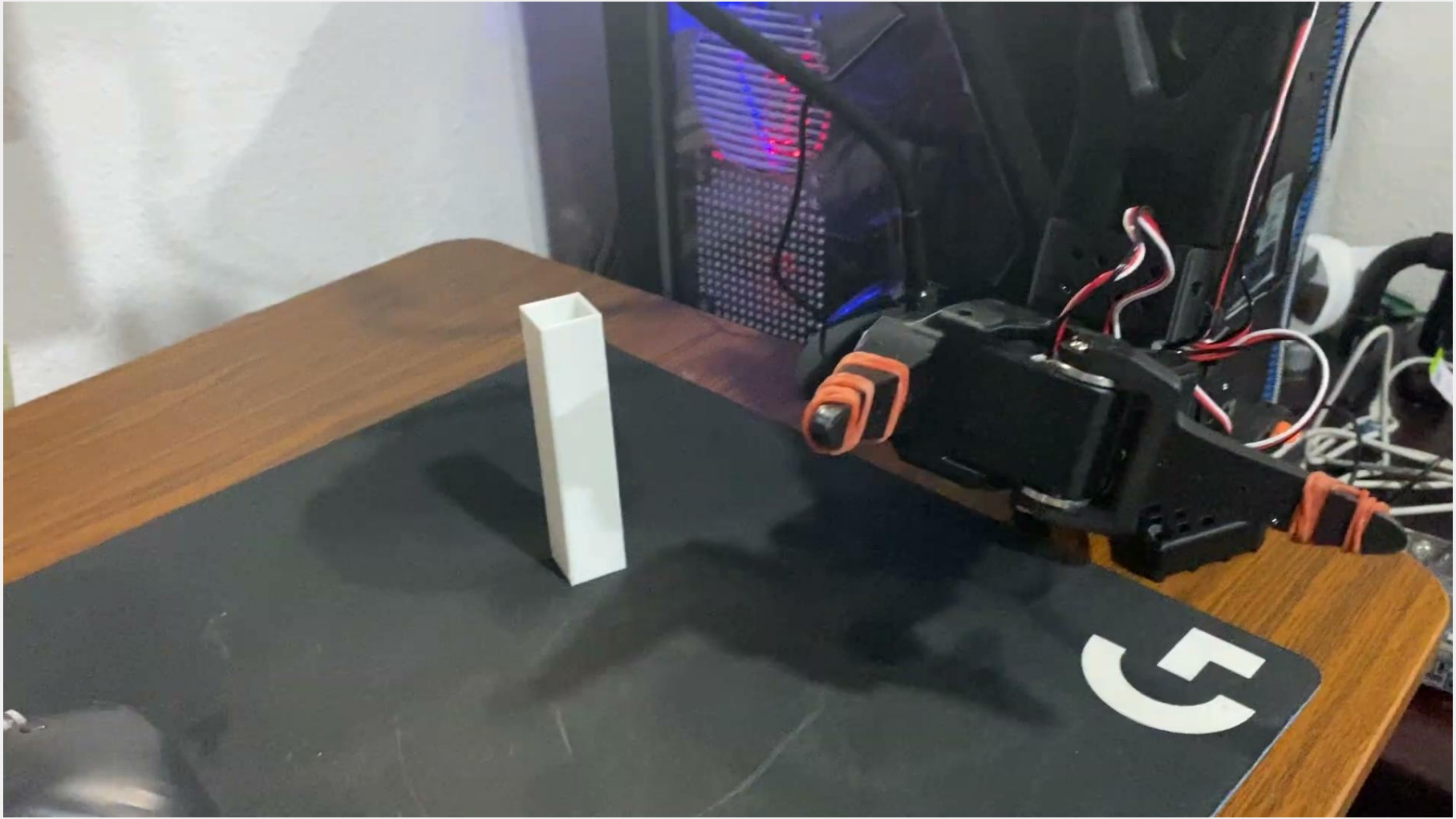
Implement reinforcement learning approach

For more consistent pouring motion and further minimizing spillage

03

Implement liquid level detection

This will help for more precise pouring and stop when the container is full or the liquid has run out



CREDITS: This presentation template was created by **Slidesgo**, and includes icons, infographics & images by **Freepik**

Thank you