Gaze-Powered CoBot

Smart Robotics Final Project

Pavone & Salici, 2024

Problem…

Images are taken from the web. Credit to the respective authors

...Goals

Seamless Human-Robot Interaction

Facilitate user interaction by placing objects near the user wearing smart glasses.

Automate Pick & Place Operations

Use motion planning for precise robotic movement to pick up objects from locations detected using markers.

Real-Time Position Communication

Accurately communicate the position of boxes relative to the collaborative robot.

No need for robot vision

Ensure real-time world position estimation using Aruco Marker and smart glasses.

Devices

Images are taken from the web. Credit to the respective authors

Project Aria

Meta project

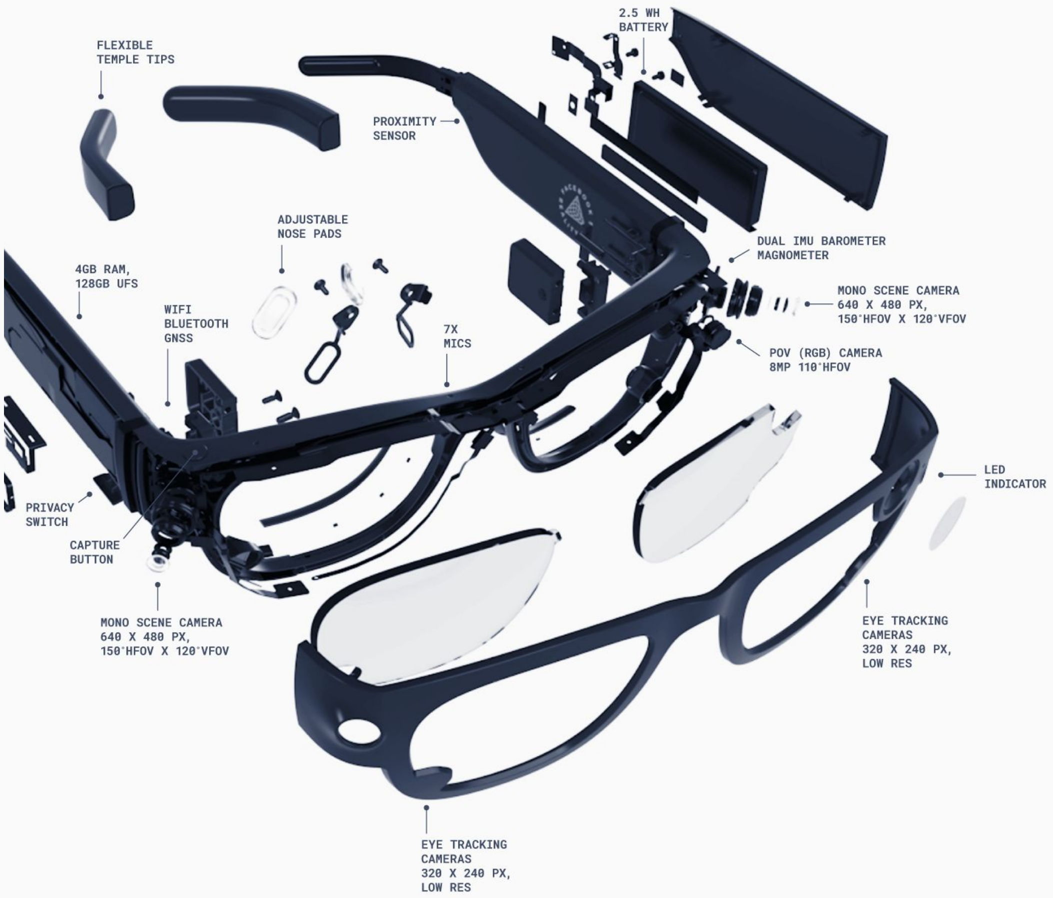
AR research

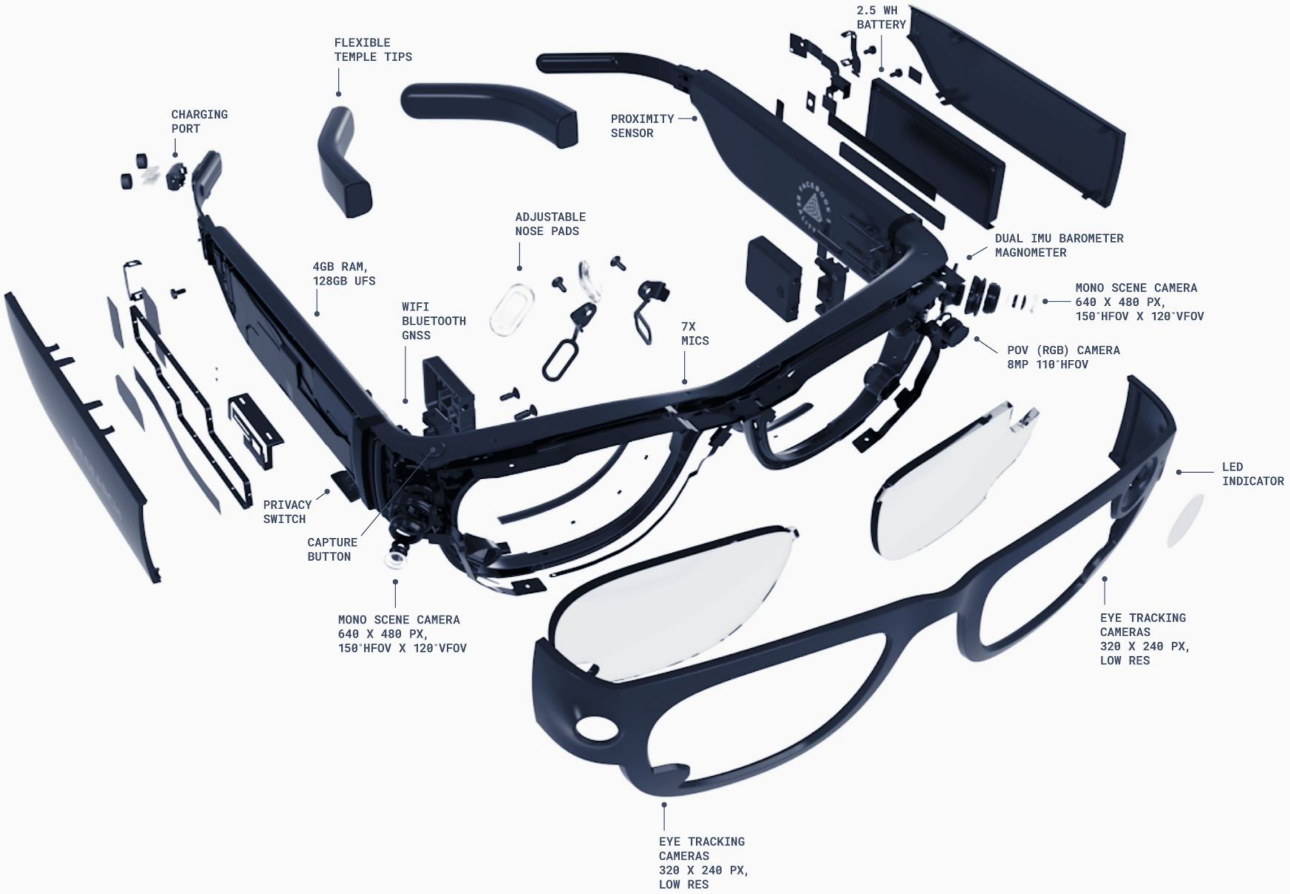
5 cameras

7 mics

2 IMU

...much more!





Project Aria Glasses

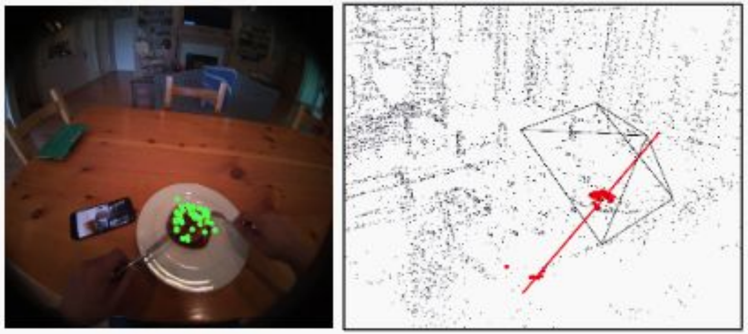


Figure 10. Eye gaze computed on a recording in which the user is looking at the food on their plate. Left: The RGB image with semi-dense points that are close to the gaze ray projected as green dots. Right: The Project Aria device pose (shown as RGB camera frustum), semi-dense points, and eye gaze ray. Semi-dense points close to the gaze ray are highlighted in red.

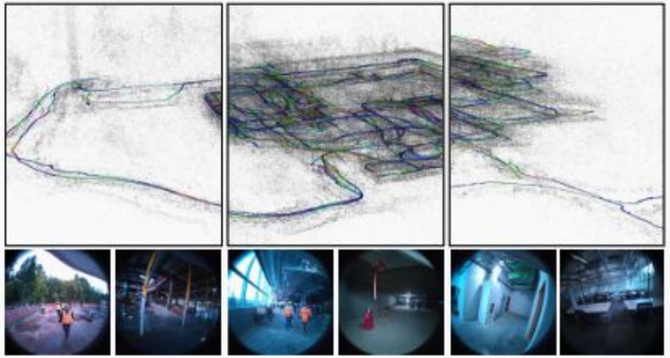
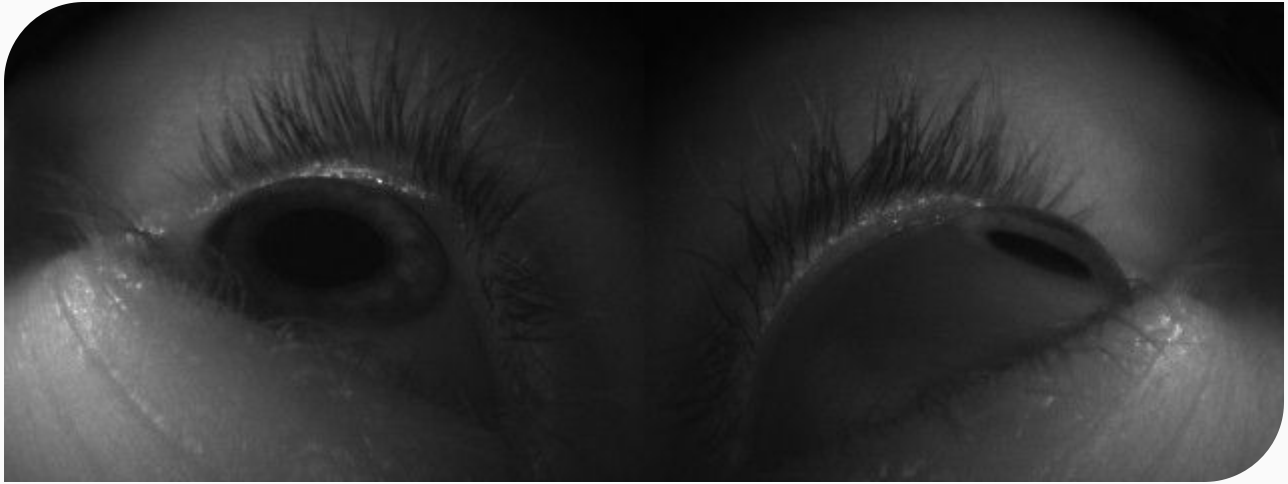
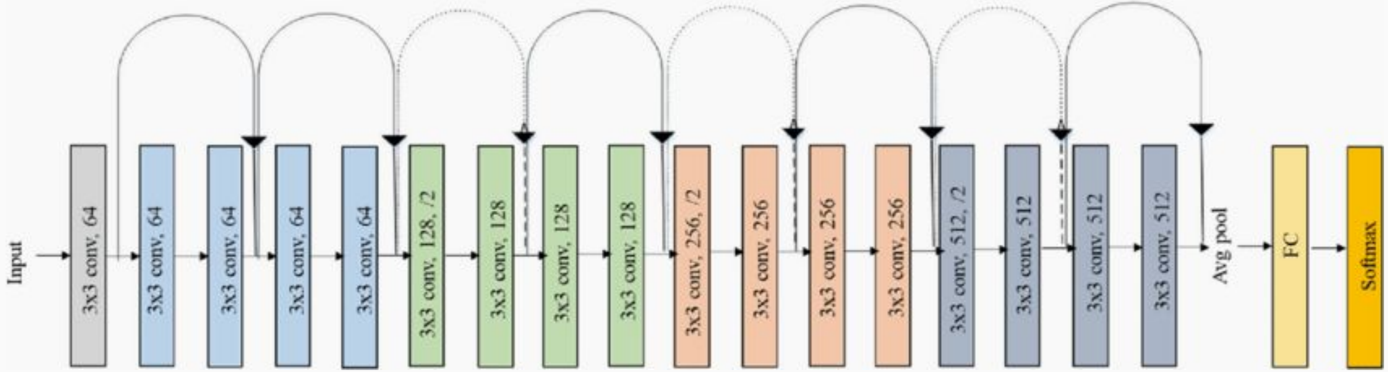


Figure 11. Top: A 3D map created from 275 Aria recordings (175 hours of data), captured over 15 months in a construction site, showing – from left to right – the state of the map after one, six, and fifteen months. The images at the bottom are samples from the respective points in time, depicting the transformation from an empty lot to an office building.





Eye Tracking Cameras Frame - Social Eye model base architecture

Franka Emika Panda

Degrees of freedom: 7

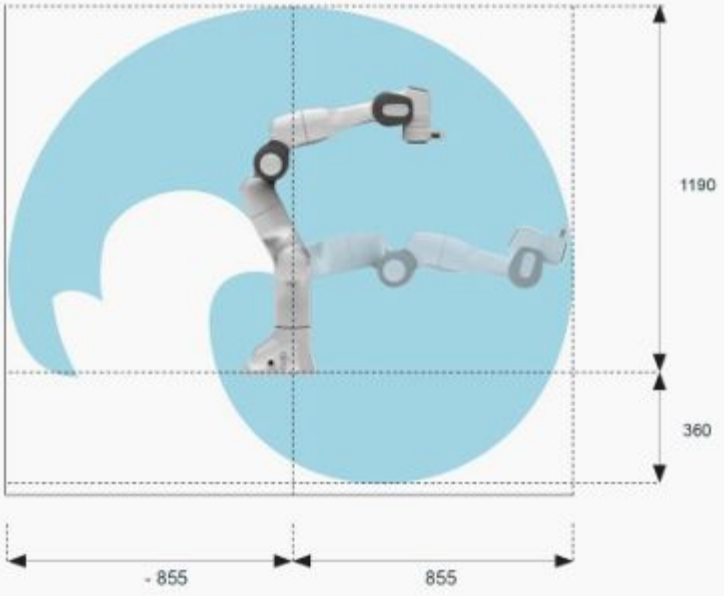
Payload: 3kg

855 mm maximum reach

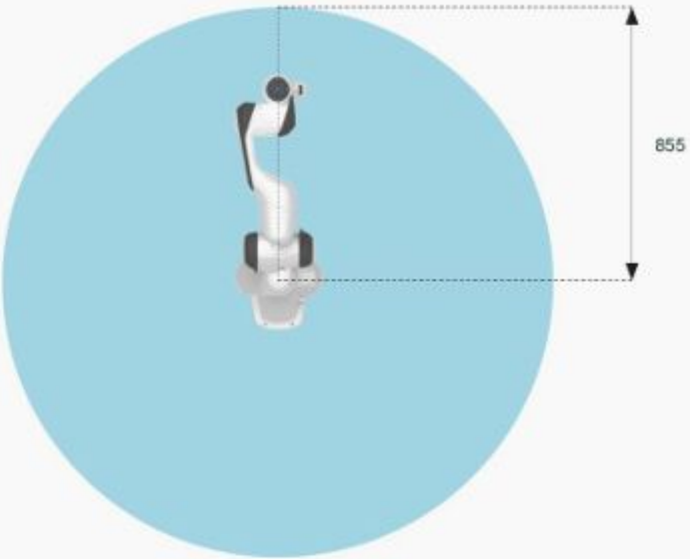
F/T Sensing link-side torque sensors in all 7 axes

Cartesian velocity limits: 2m/s

Joint velocity limits: 150 or 180 °/s



1. Arm workspace side view



2. Arm workspace top view

Panda Workspace

Software Architecture

ROS Control

franka_hw

Interfaces

- Model
- State
- Joint position
- Joint velocity
- Cartesian pose
- Cartesian velocity
- Torques

franka_control
srv + action_lib

franka_gazebo

Interfaces

- Model
- State
- Torques

franka_hw_sim
franka_gripper_sim

franka_gripper
action_lib

franka_description
URDF + Meshes

franka_visualization

panda_moveit_config

MoveIt!

project_aria_utils

python_bridge
(socket client)

tailscale

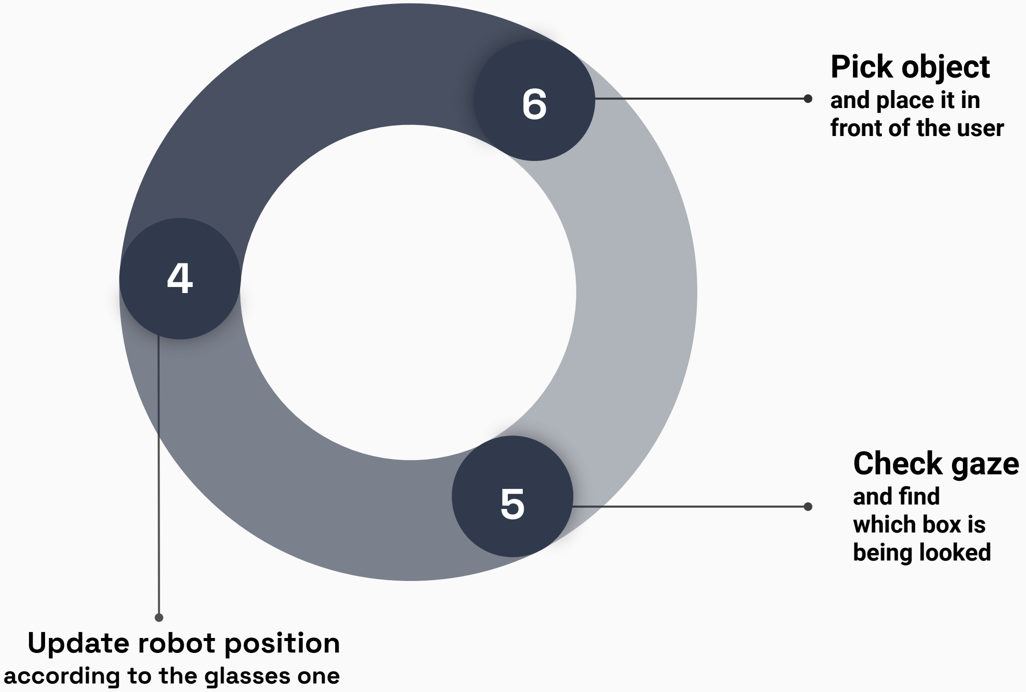
Application lifecycle

Once in a session

1. Gazebo, Aria Project, ROS init
2. Glasses detect boxes position
3. Word Frame is calculated

E. Extra: Camera Undistortion

Realtime



$$ {}^{\mathrm{cobot}}\mathbf{p}_{\mathrm{cobot}} = \mathbf{0} $$

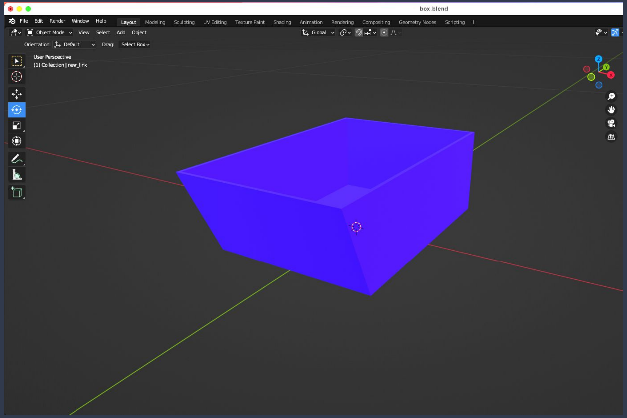
$$ {}^{\mathrm{cobot}}\mathbf{p}_{\mathrm{glasses}} = {}^{\mathrm{glasses}}[\mathbf{R}|\mathbf{t}]^{-1}_{\mathrm{cobot}} \times {}^{\mathrm{glasses}}\mathbf{p}_{\mathrm{cobot}} $$

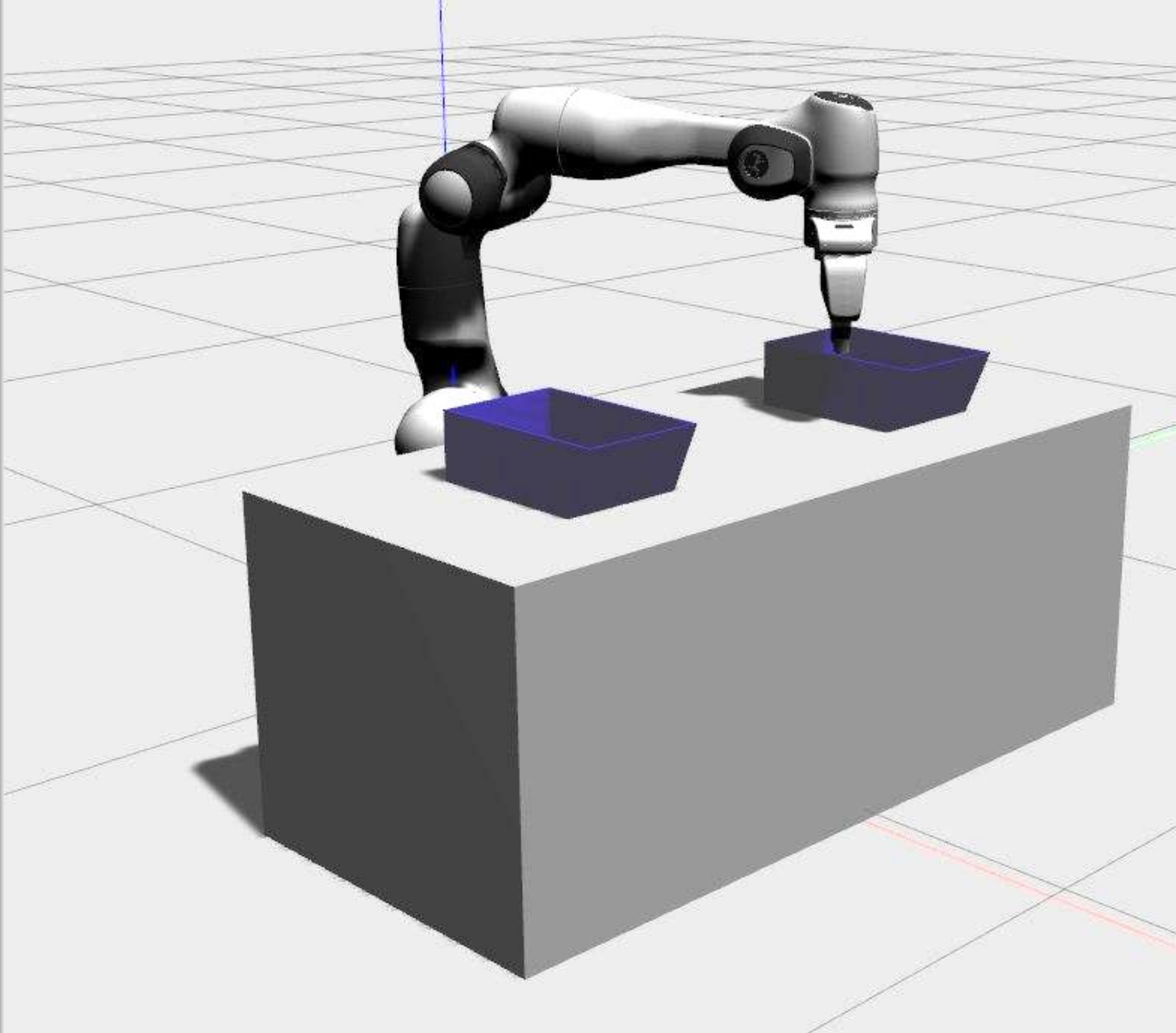
$$ {}^{\mathrm{cobot}}\mathbf{p}_{\mathrm{box[i]}} = {}^{\mathrm{glasses}}[\mathbf{R}|\mathbf{t}]^{-1}_{\mathrm{cobot}} \times {}^{\mathrm{glasses}}\mathbf{p}_{\mathrm{box[i]}} $$

Position transformation in the real world frame (i.e. cobot frame)

Gazebo environment

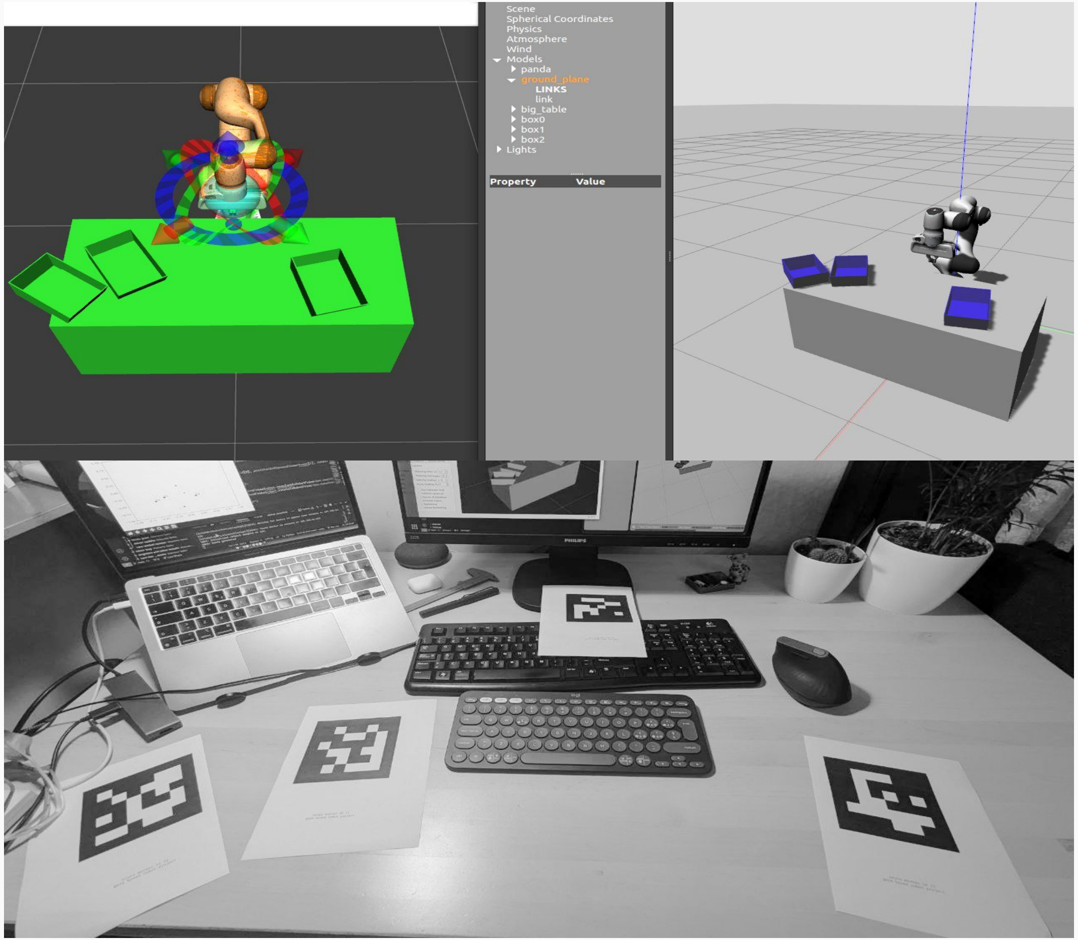
- Custom Blender Box Model





Gazebo environment

- Dynamic box position



Robot Localization

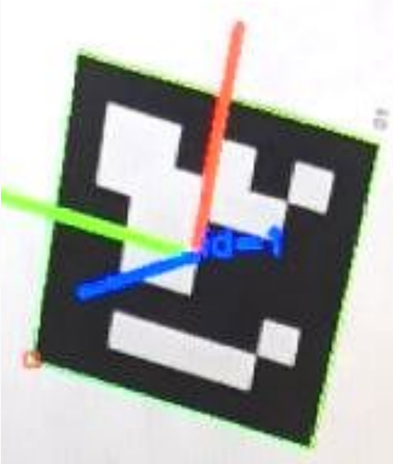
Aruco markers: Square binary markers used for detecting, localizing and identifying robot and the boxes.

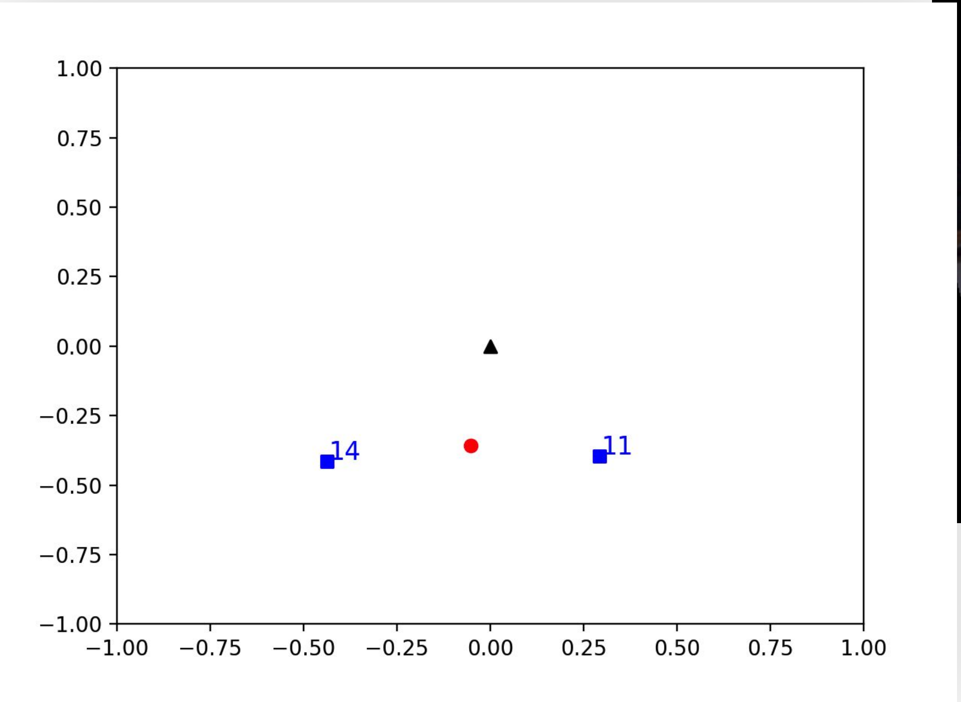
- 6x6_250 Aruco dictionary

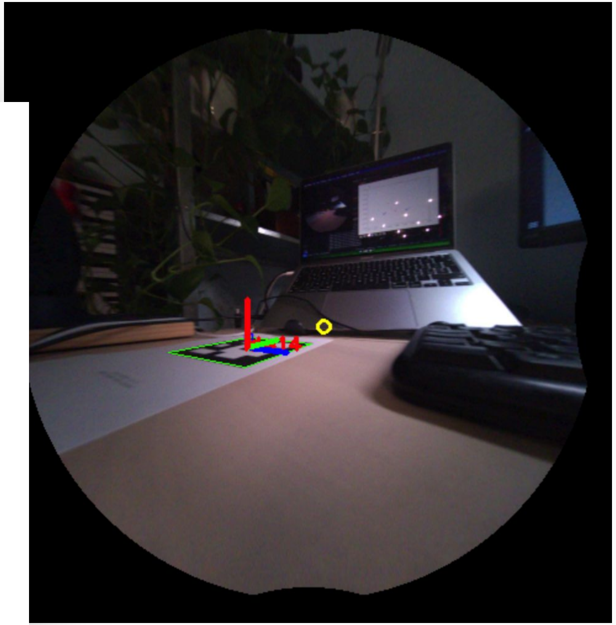
```
corners, ids, _ =
cv2.aruco.detectMarkers(image)

success, rvec, tvec =
cv2.solvePnP(object_points, corners[i],
camera_matrix, camera_distorsion)
```

OpenCV







Real time 2d virtualization and Aria Project RGB camera video stream frame

MoveIt

MoveIt is a powerful motion planning framework integrated with ROS.

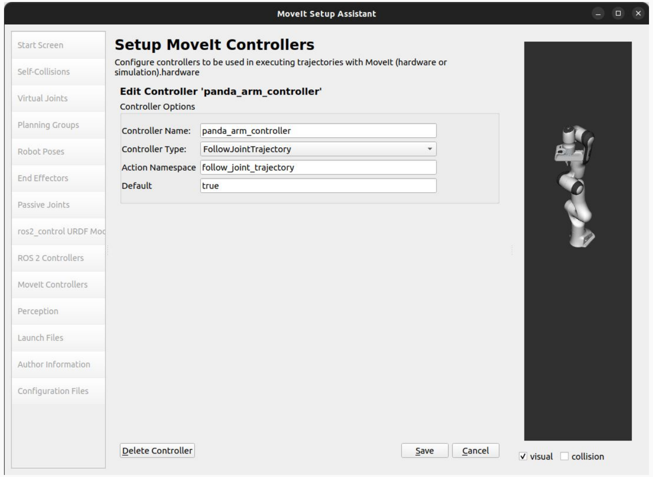
We utilized **MoveIt's Python APIs** to handle high-level tasks such as kinematics and path planning.

It enables control over join movements, cartesian coordinates and motion planning.

MoveIt

KDL kinematics plugin

- default **inverse kinematics (IK)**
- Orocos KDL package.
- computes the **joint positions to achieve a target end-effector** pose (while respecting joint and safety limits defined in the URDF)

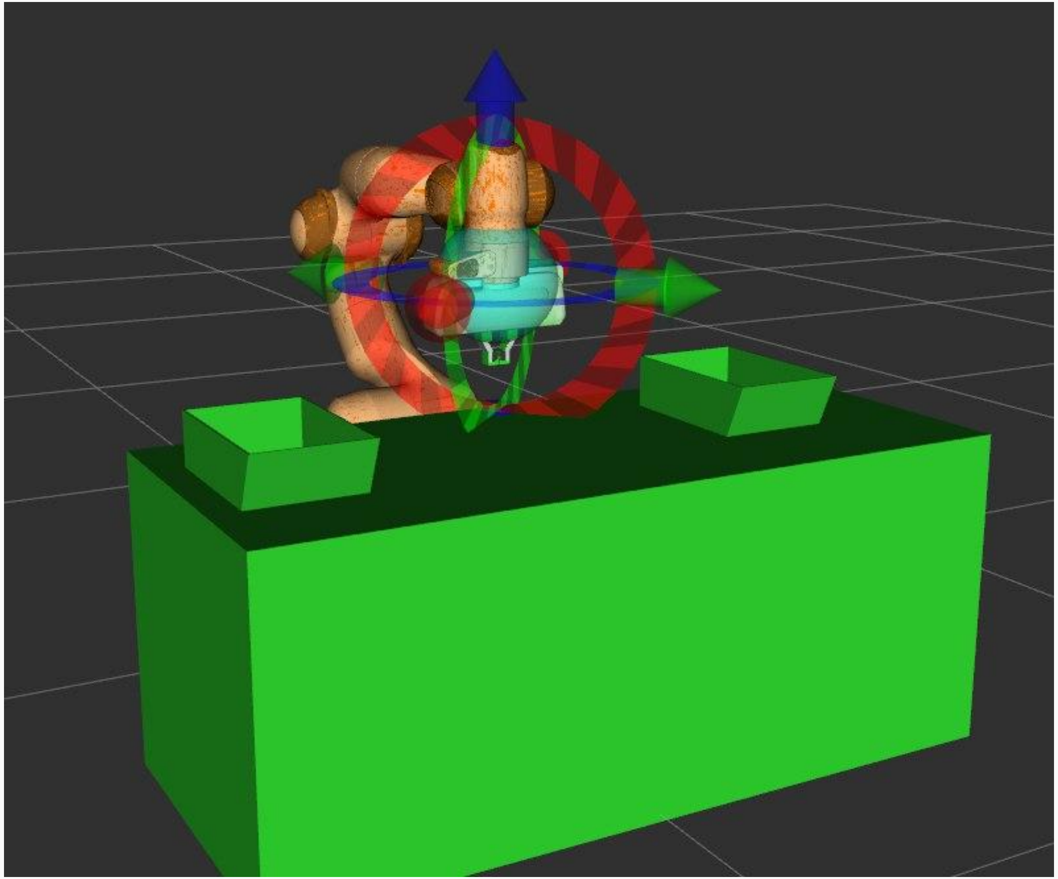


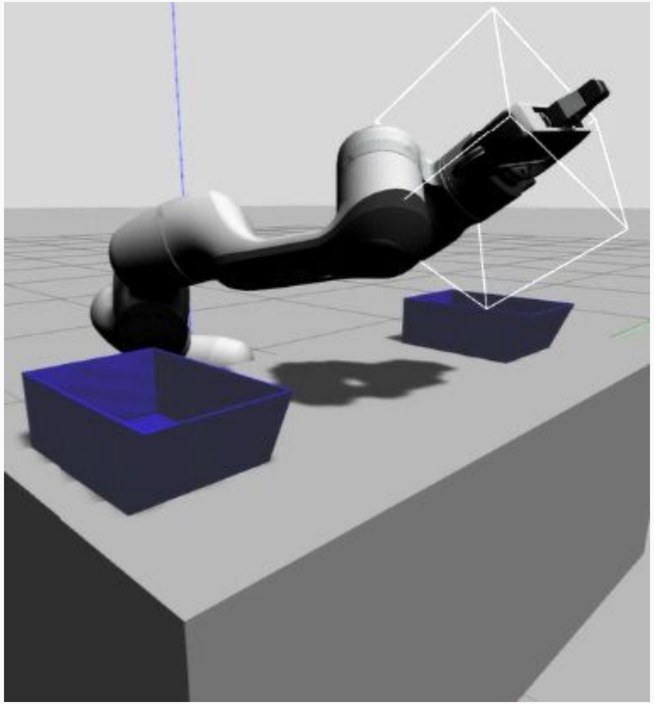
Obstacle avoidance

MoveIt's ***PlanningSceneInterface***:

- to add static and dynamic objects for obstacle avoidance during motion planning
- Static objects are set at the start, while boxes are only considered during motion.
- Using the OMPL library, with RRTConnect as the default planner, MoveIt generates collision-free trajectories, balancing speed and accuracy for real-time tasks like pick and place.

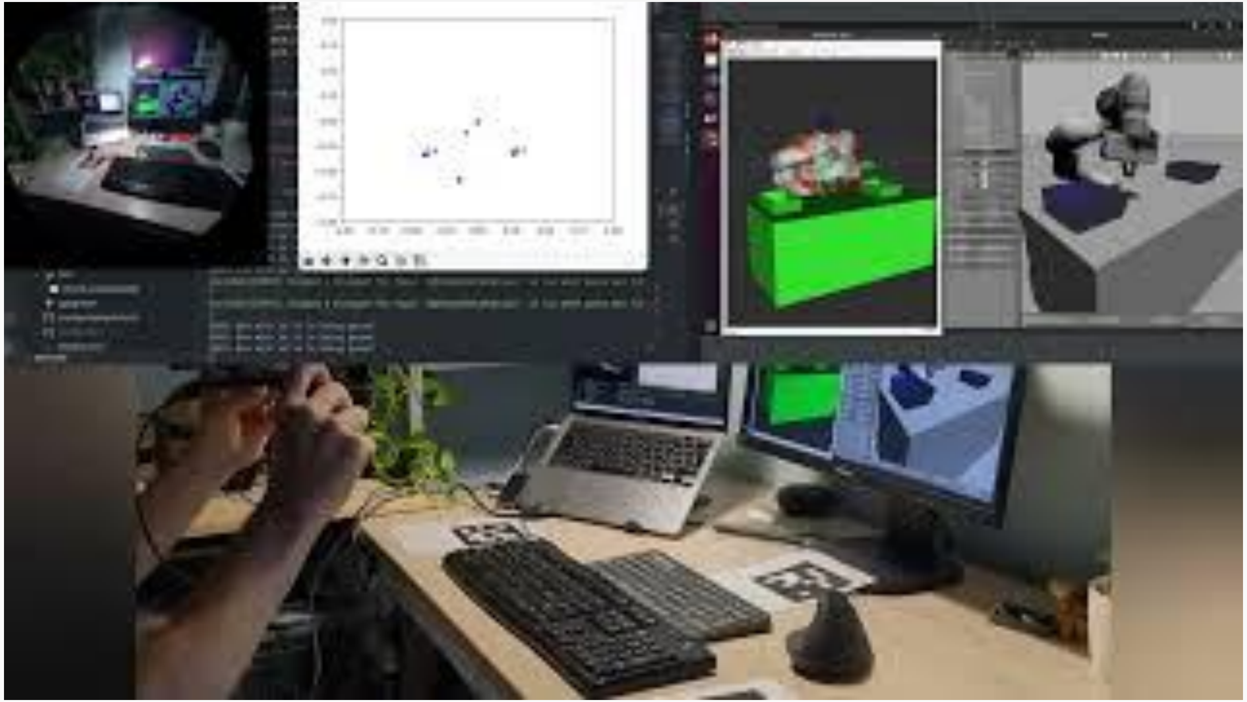
MoveIt





User position can be used as target position (impedance control for safety)

Demo



Thanks for your attention

Gaze-Powered CoBot Project
Cosimo Pavone, Giacomo Salici
{272636, 270385}@studenti.unimore.it
Course of Smart Robotics, a.y. 2023-2024

https://github.com/jacksalici/gaze-robot-assistant