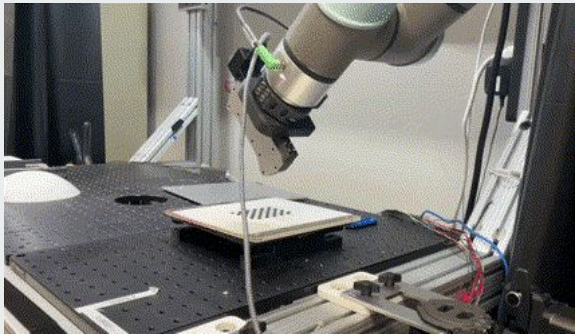
Easy Calibration Toolbox

Fujun Ruan, Letian Leng, Siyu Zhou, Yibo Chen



Overview

- Building a toolbox to increase the **efficiency** and **accuracy** during calibration
- Providing high-quality calibration samples
- Offering sample inspection methods
- Calibration include:
  - Intrinsics Calibration
  - Hand-eye Calibration
  - Camera-laser extrinsics Calibration
  - ……

Hardware/software setup

- Hardware
  - Camera sensor
  - UR5e Robotic Arm
- Software
  - Environments - Ubuntu 20.04/18.04
  - Published Calibration package
  - ROS1 - for msg communication
  - MoveIt- for low level planning & control
- Installation
  - Simple shell script
    - Camera driver
    - ROS
    - MOVEIT

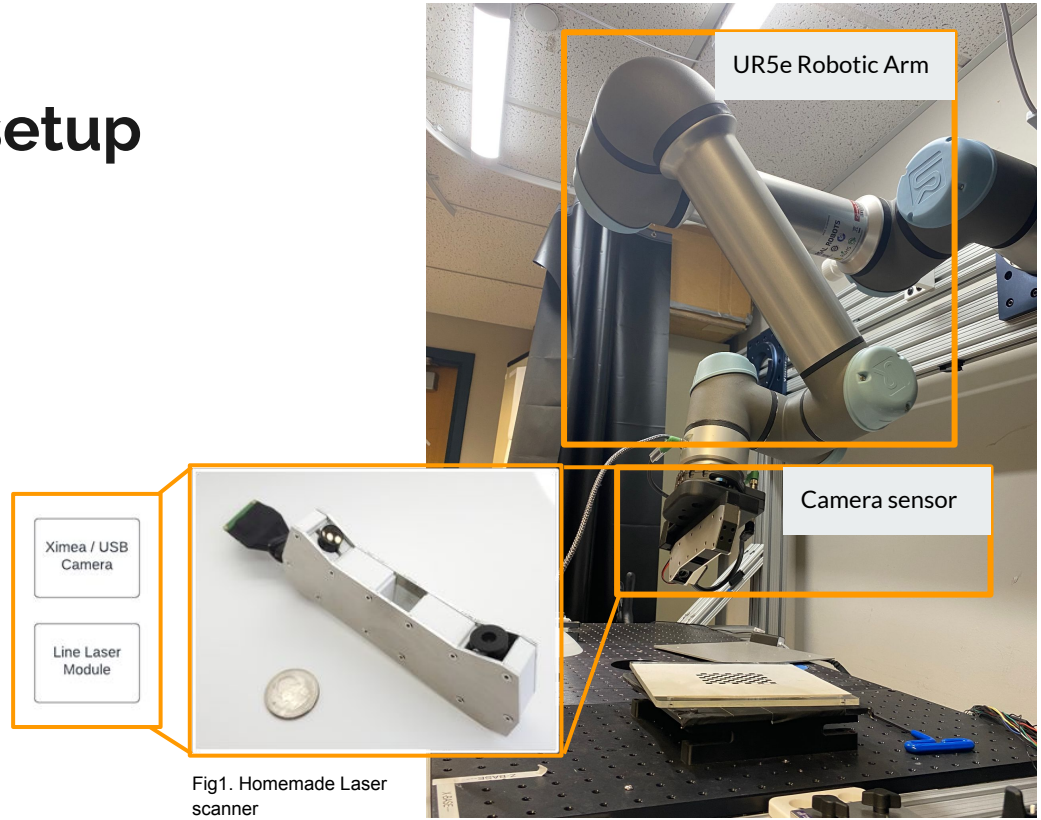
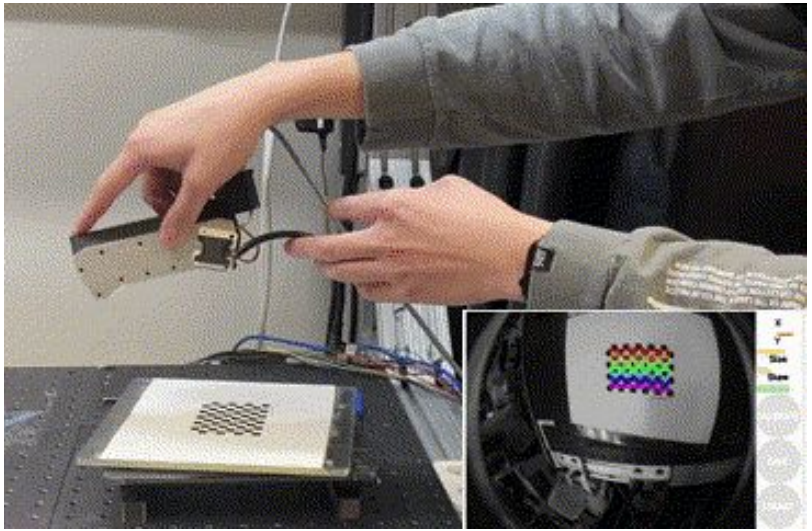


Fig1. Homemade Laser scanner

Camera Intrinsics Calibration

- Critical to rectify image distortion
- Existing intrinsic calibration toolboxes
  - Blur images
  - Can't cover enough area



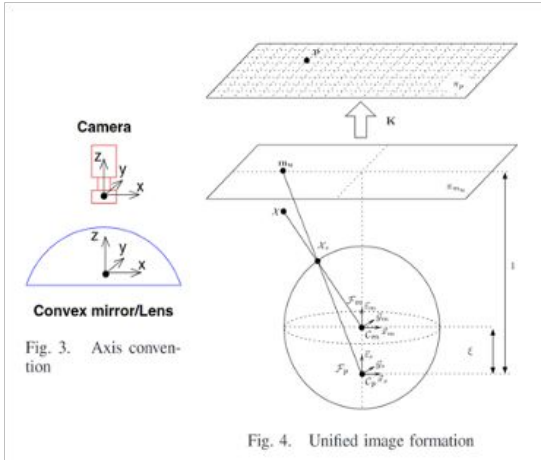


Fig. 3. Axis convention

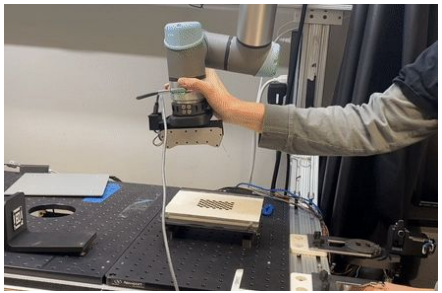
Fig. 4. Unified image formation

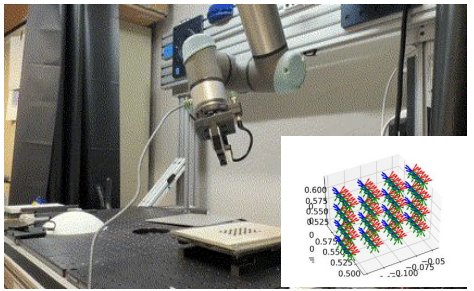
Improvement in finding the undistortion model

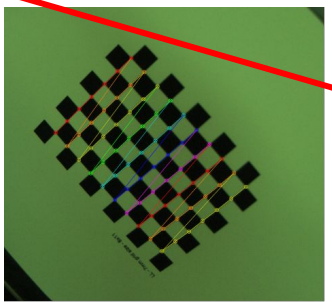
Challenge: Manual picture collection

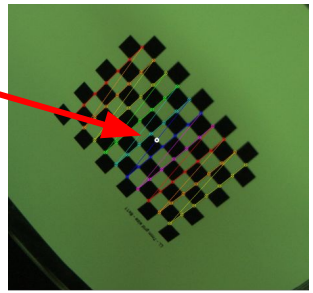
Solution: Adding automation path planning in picture collection process

1. Mei model
2. Robot arm -> center position + visual feedback
3. Plan the trajectory of the calibration







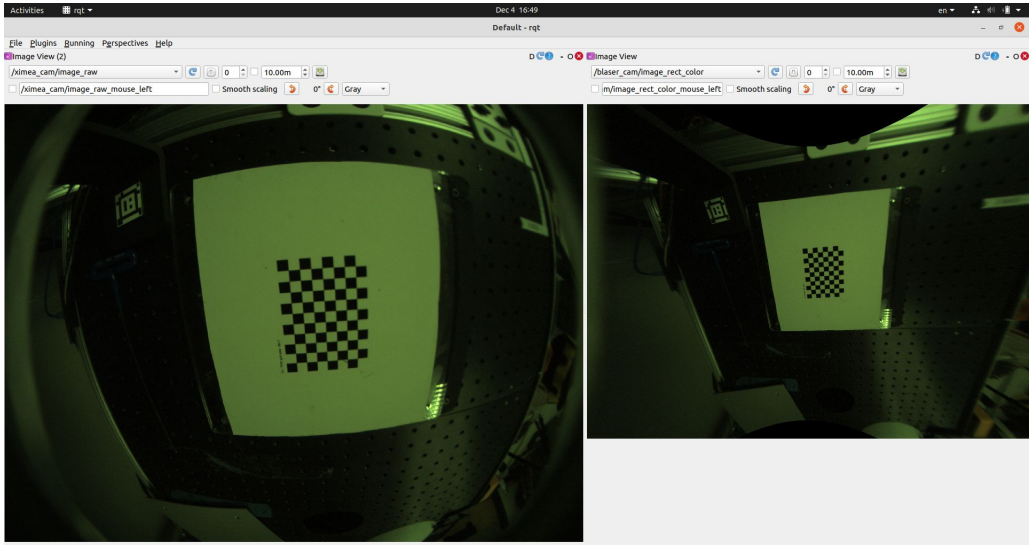


Improvement in finding the undistortion model

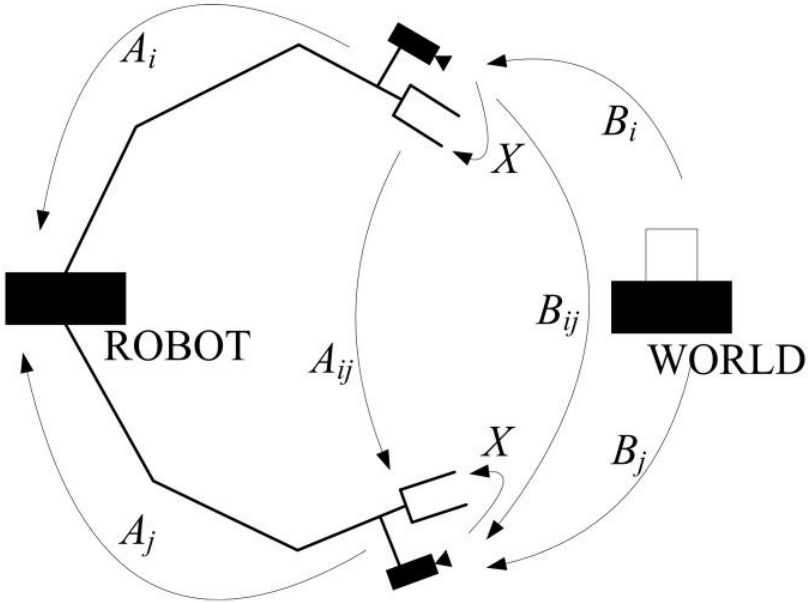
Challenge: Manual picture collection

Solution: Adding automation path planning in picture collection process

1. Mei model
2. Robot arm -> center position + visual feedback
3. Plan the trajectory of the calibration



Hand-eye calibration (traditional)



Hand–eye calibration equation can be represented as:

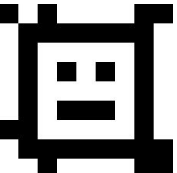
$$A_{ij}X = XB_{ij}$$

Improved Hand-eye calibration

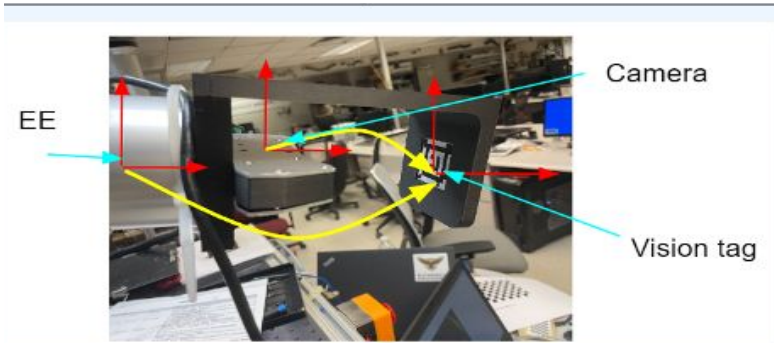
1. Hardware setup
2. Use CAD tools to get transform matrix from end effector to the vision tag
3. Use ximea camera node and read transform matrix(from camera to vision tag) in python script (handeye.py)
4. Use the script to calculate the transform matrix(from end effector to camera)
5. Result check

Advantage:

- Don’t need hundreds of pictures
- April tag provides more accurate result

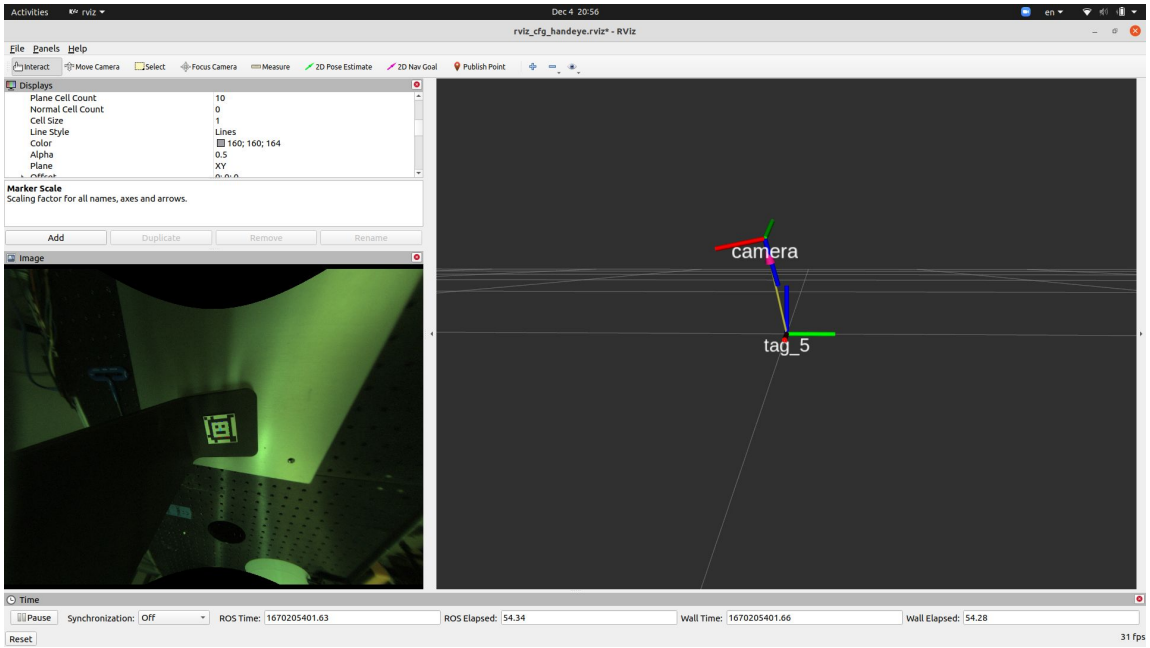


April Tag Figure

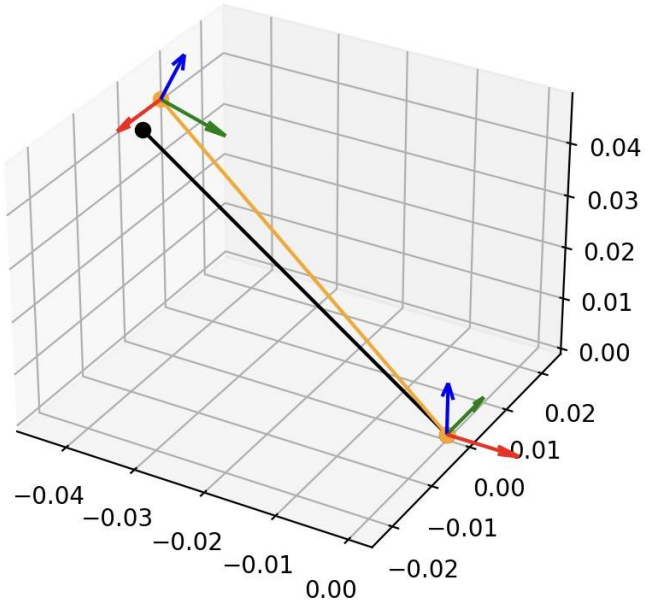


Hardware setup

Hand-eye calibration interface



Hand-eye calibration result



The equation for calculation is:

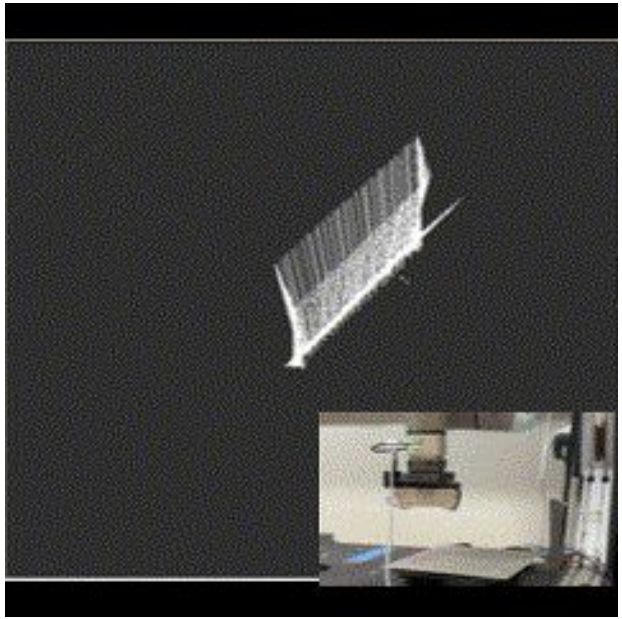
$$M = M_2 M_1^{-1}$$

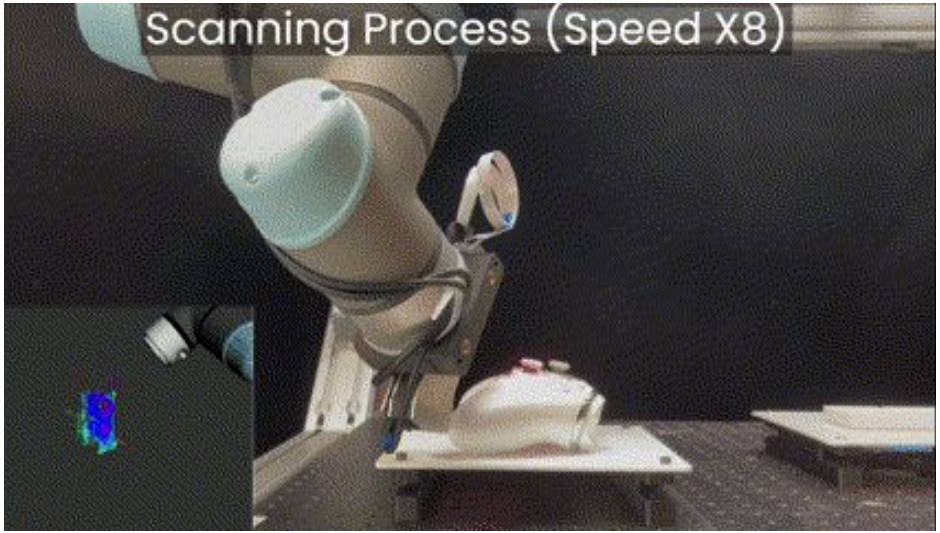
where M is the transfer matrix from end effector to camera,
M1 is transfer matrix from camera to vision tag,
M2 is transfer matrix from end effector to the vision tag.

red arrow represents the x-axis
green arrow represents the y-axis
blue arrow represents the z-axis

Black line represents the estimated position and orange line represents the calculated position

Bad calibration vs. Good calibration



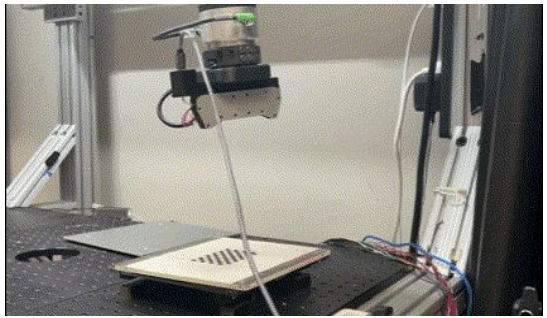


Failed scanning process demo(left) and successful scanning process demo(right)

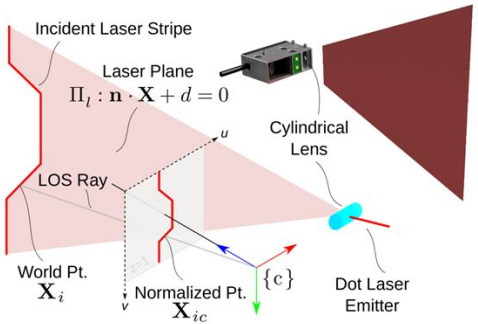
Camera-laser Extrinsics Calibration

Challenge: Manual picture collection (traditional procedure)

Solution: Automation image-collection process, while filtering out noise image

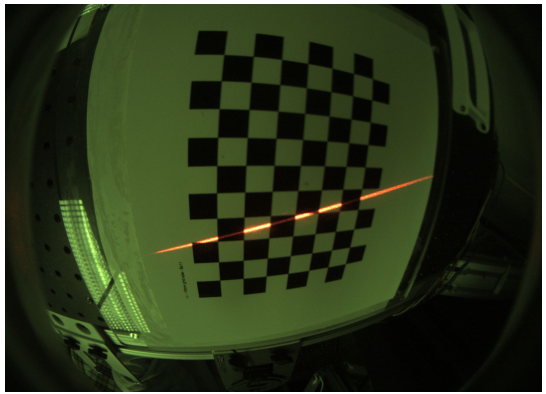


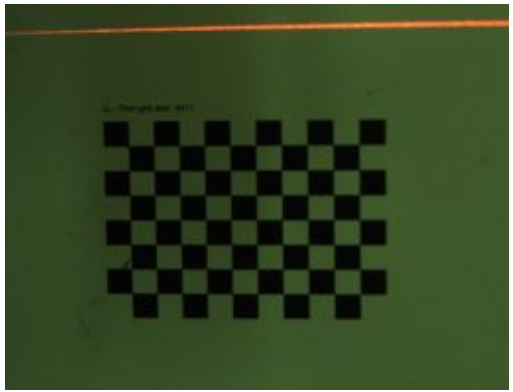
[1]



$$\begin{cases} \vec{n} \cdot \mathbf{X} + d = 0 \\ \mathbf{X} = t \cdot \pi_c^{-1}(\mathbf{x}) \end{cases} \Rightarrow \mathbf{X} = \frac{-d}{\vec{n} \cdot \pi_c^{-1}(\mathbf{x})} \cdot \pi_c^{-1}(\mathbf{x})$$

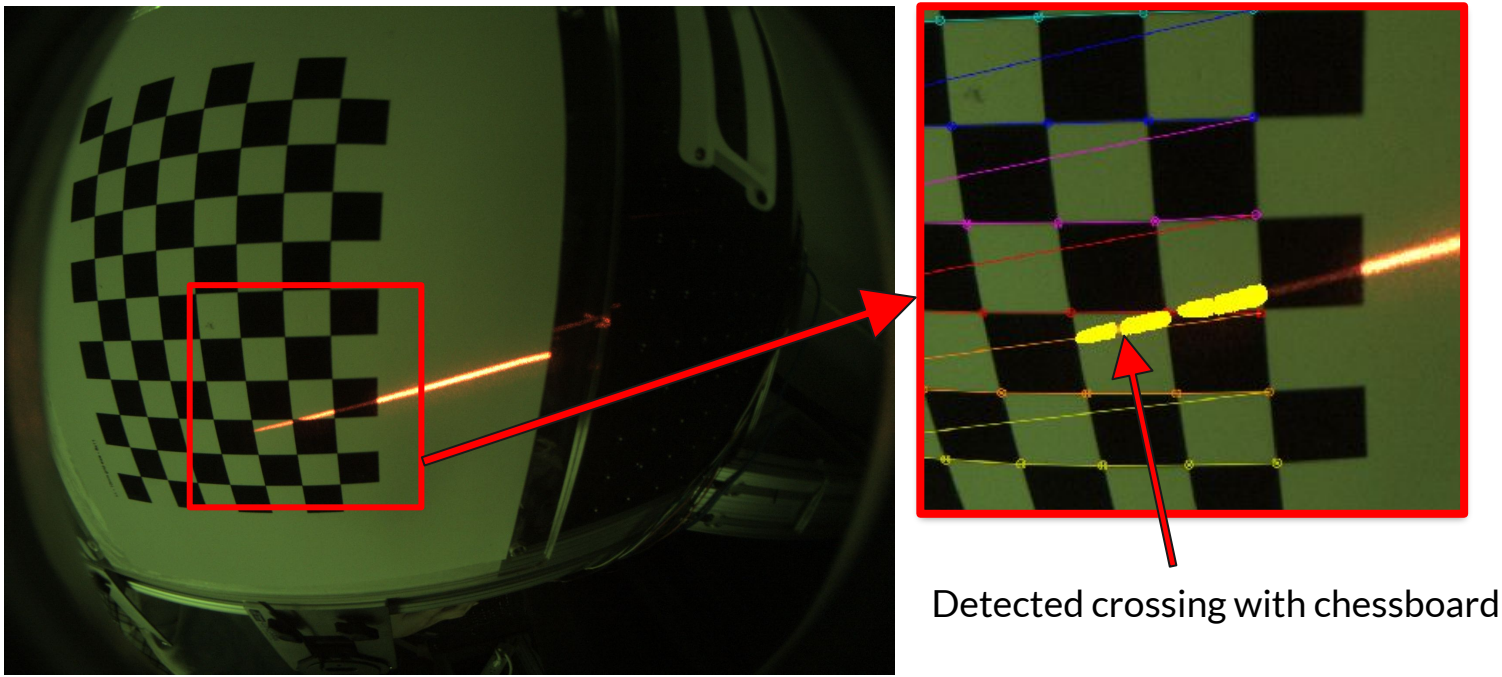
Two constraints





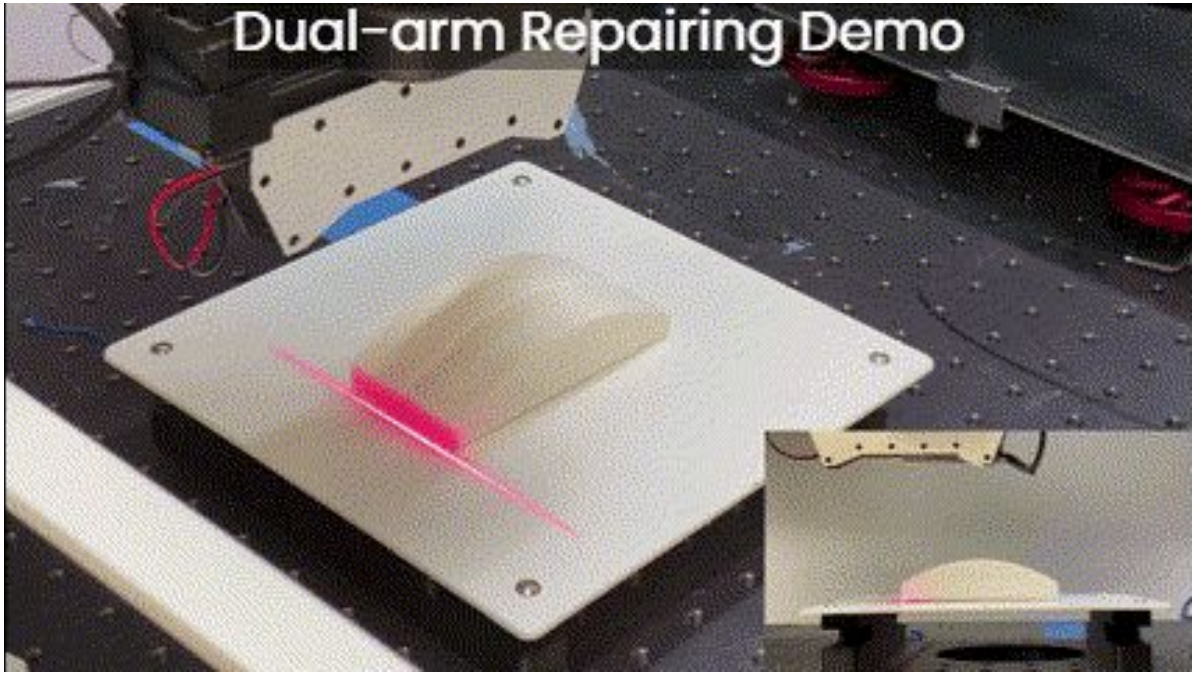
[1] Visual-Laser-Inertial SLAM Using a Compact 3D Scanner for Confined Space. Cheng et. al. ICRA 2021

Camera-laser Extrinsics Calibration Result (image-filter)

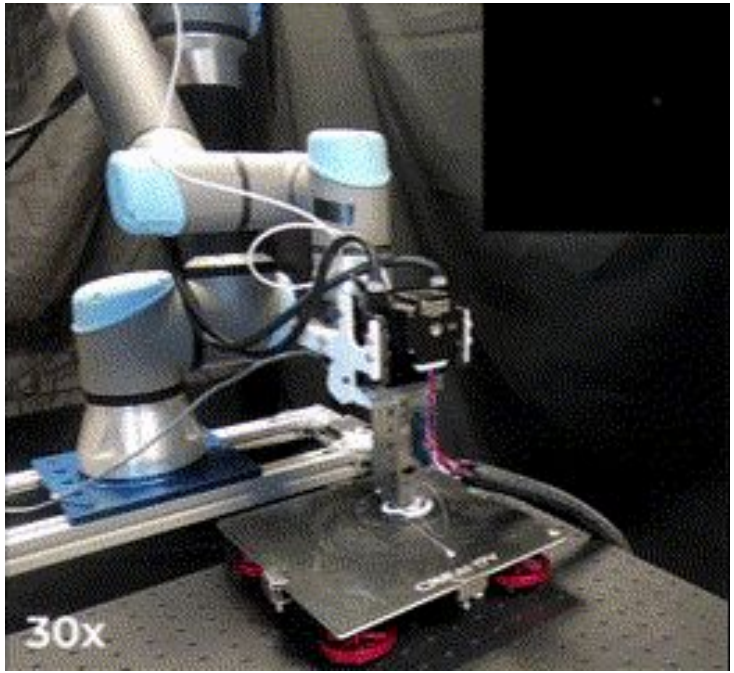


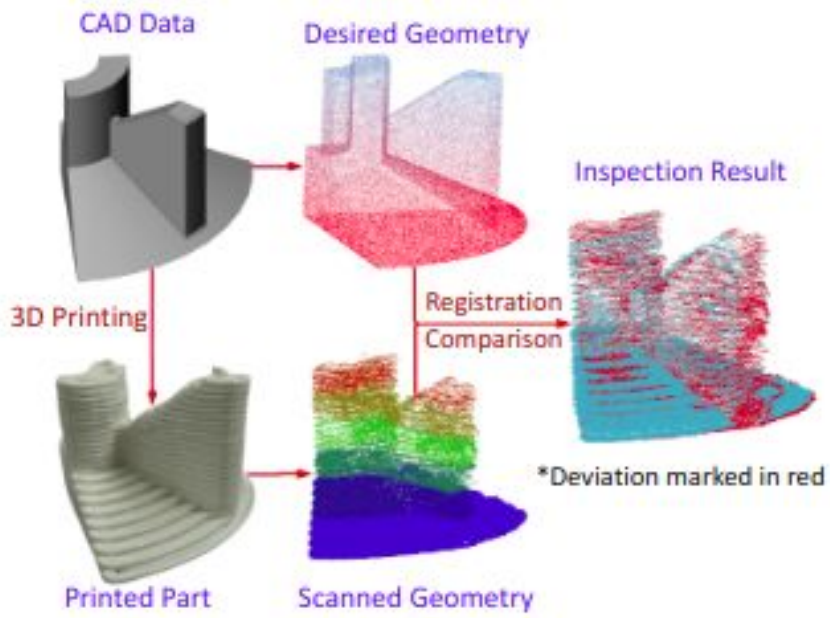
Detected crossing with chessboard

Calibration result application



Calibration result application





Future Work

- UI for user input value such as IP of the robot (using RVIZ)
- Better method for evaluating the result
- Can use adapt with different input image message and robot arm