Hand-eye calibration

Umberto Castellani

Robotics, vision and control

Overall aim

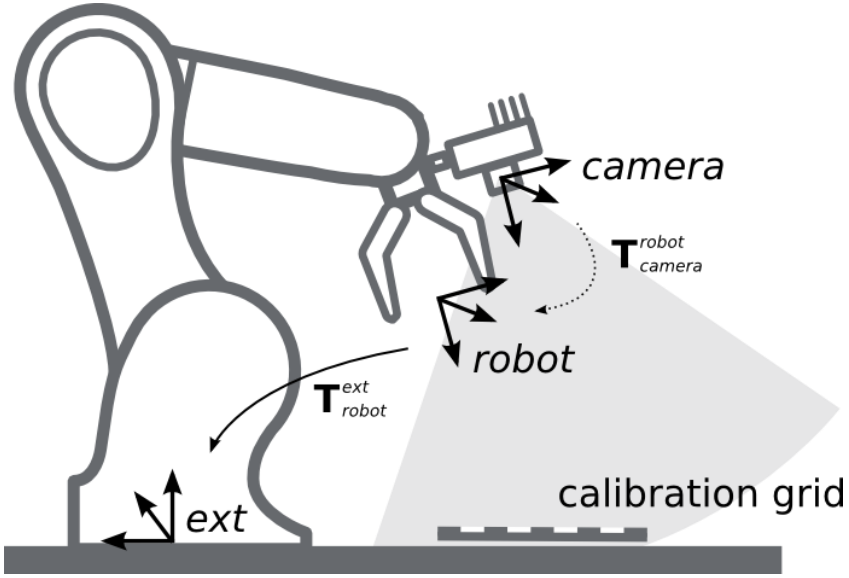
The overall aim of **hand-eye calibration** procedure is to find the **camera-to-robot** transformation.

In the basic scenarios:

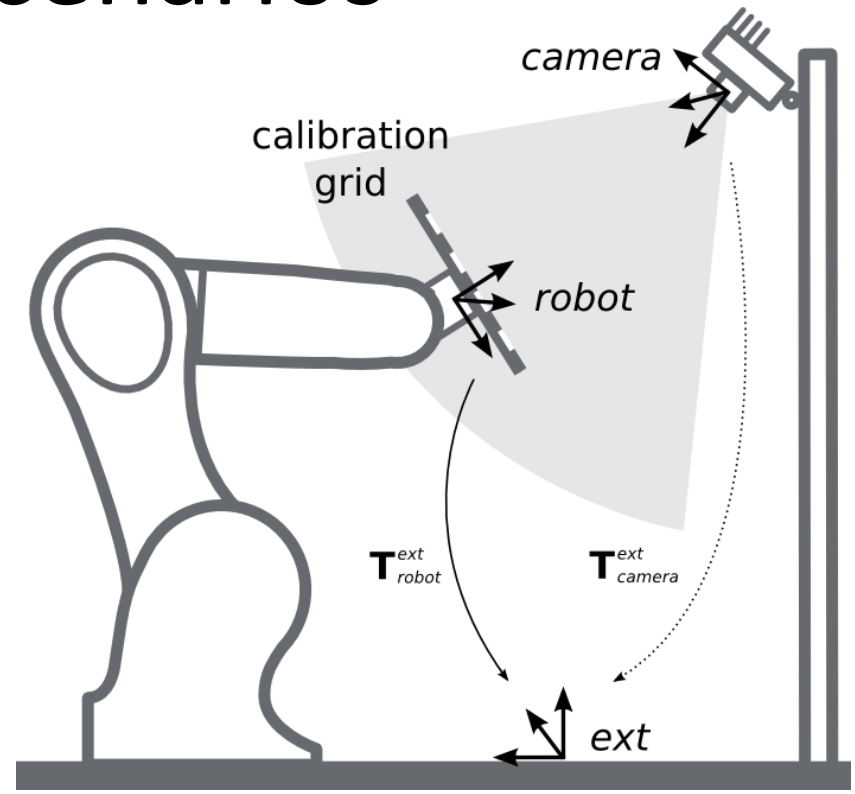
- The scene is composted by one object, one camera (e.g., RGBD device), and a robot arm with a gripper,
- The camera observes the object in the scene, detect the object position and tell the robot where to pick it.

The camera has to know the robot positon to transfrom the object coordinates from the camera to the robot reference system !

Different scenarios



camera attached to the end-effector



static camera

https://answers.opencv.org/answers/204935/revisions/

System calibration

A calibration object is used

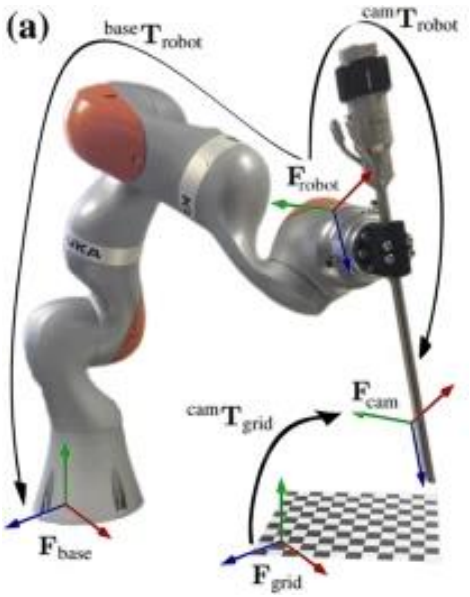
- The aim of calibration object is to easily detect 3D-2D correspondences,
- Sometimes we ask the end-effector to touch the interest points of the calibration object,
- The calibration object may define the world reference system.

Calibration System

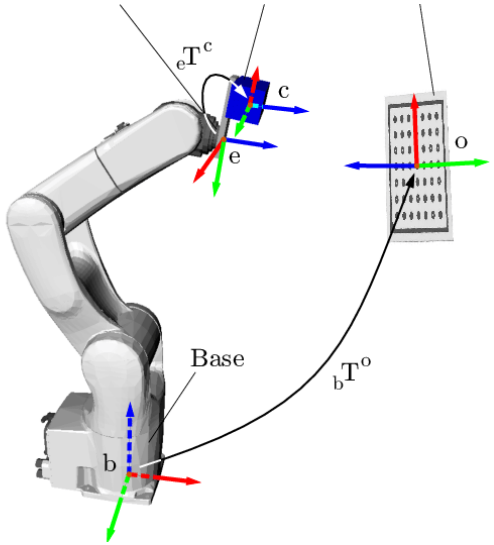
Different calibration scenarios:

- Chessboard mounted on the end-effector (we need to estimate also the transformation between the end-effector and the chessboard).
- Fixed Chessboard with camera mounted to the end effector (we need to estimate also the transformation betwen end-effector and camera)

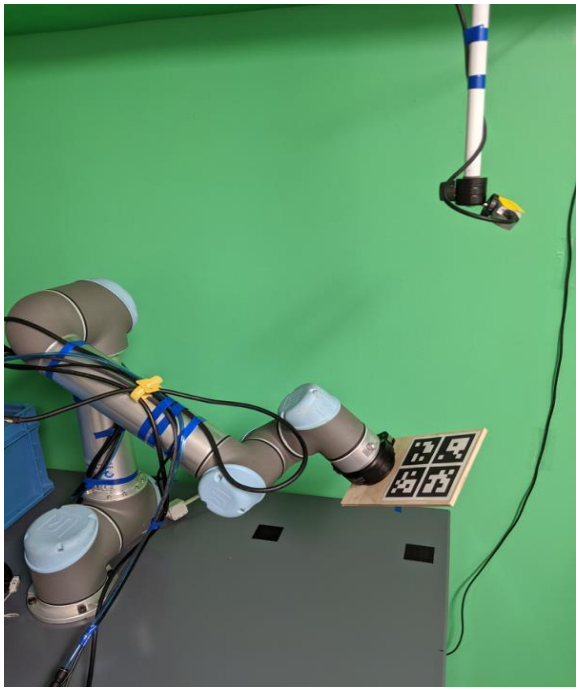
Calibration system



End-effector touches the calibration object



Fixed calibration object and camera mounted to the end-effector



Fixed camera and calibration object mounted to the end-effector

Involved transformation

- For each scenario we define the respective **transformations chain,** for example in the fixed camera scenario**:**

$$p\_cam = K * BaseT * armPose * gripT * P\_ch$$

where

- p_cam= pixel coordinates,
- K =camera projection matrix,
- BaseT= robot to camera transform,
- armPose=end-effector to robot transform,
- gripT=chessboard to end-effector transform,
- *P_ch= 3D coordinates of points on the chessboard

Involved transformation

- For each scenario we define the respective **transformation chain,** for example**:**

$$p\_cam = K * BaseT * armPose * gripT * P\_ch$$

where

- p_cam= pixel coordinates (observed by image processing),
- K =camera projection matrix (estimates as pre-processing),
- BaseT= robot to camera transform (unknown),
- armPose=end-effector to robot transform (given),
- gripT=chessboard to end-effector transform (unknown),
- *P_ch= 3D coordinates of points on the chessboard (given).

Probem solution

- We need to combine the various trasformations in order to obtain reasonable equations to estimate our unknowns,
- We need to define a suitable optimization problem,
  - We should solve the optimization problem in closed-form, or...
  - We should employ an iterative (numerical) method.

The litterature on hand-eye calibration differs on the definition of the optimization problem according to the involved Transformations (and the respective scenarios)

Homework

- Try to run this code:

https://github.com/ZacharyTaylor/Camera-to-Arm-Calibration

- Try to understand the main steps,
  - The most important part is the ‘ProjectError’ function.