

## LAB 7

# Robot Vision II: Camera Calibration

### 7.1 Important

Read the entire lab before starting and especially the “Grading” section so you are aware of all due dates and requirements associated with the lab. There will be only one online session for this lab, so it is important that you prepare for that one opportunity to work with your TA.

### 7.2 Scope

- Camera Calibration - Obtain the relation between camera and world reference frame
- Robot-Camera Calibration - Obtain the mapping between the camera and robot coordinates
- Image-Guided Manipulation - Plan and execute path for manipulation task

### 7.3 Objectives

This is the capstone lab of the semester and will integrate your work done in previous labs with python, ROS, and forward and inverse kinematics. It will also introduce you to some basic OpenCV techniques. In this lab you will:

- Use OpenCV functions to distinguish the shape of the blocks and find the centroid and angle of them.
- Develop transformation equations that relate pixels in the image to coordinates in the world frame

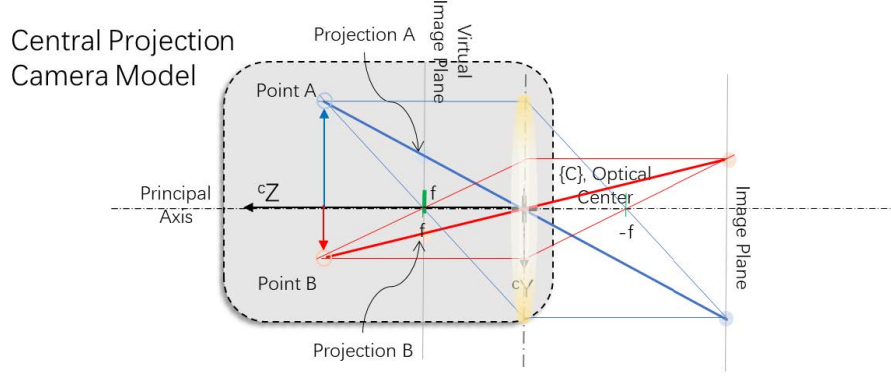


Figure 7.1: Central Projection Camera Model

- Report the world frame coordinates  $(x_w, y_w)$  of the centroid of each block in the camera's view.
- Move the block to a predefined position in the robot's work area.

## 7.4 Background

### 7.4.1 The Camera Model and Calibration Process

Robotic systems commonly rely on visual information from the surrounding to perceive the environment. The source of this information typically comes in the form of images depicting the scene of the camera views. Therefore, camera can be an extremely useful sensing device to provide a robotic system with control feedback and perception. Like many other sensors, there is usually a need to map the acquired signal (imagery) to measurement (spatial) through a calibration process based on a camera model. The central projection camera model as shown in Figure 7.1 is a representation of the image formation process relating points in world coordinates to their projection on a virtual image plane.

Camera calibration uses known image data of scene from different camera views to establish a mapping between the camera pose and the world reference. This mapping is usually in the form of a camera calibration matrix expressed as

$$M = C(R \mid t) \quad (7.1)$$

where  $R$  and  $t$  are the 3-by-3 rotation matrix and 3-by-1 translation vector, respectively, from the camera's viewpoint. Together they are known as the extrinsic matrix. Matrix  $C$  is a 3-by-3 matrix with intrinsic parameters including

its focal length  $(f_x, f_y)^T$ , principal point  $(i_c, j_c)^T$  expressed as

$$\mathbf{C} = \begin{pmatrix} f_x & a & i_o \\ 0 & f_y & j_o \\ 0 & 0 & 1 \end{pmatrix} \quad (7.2)$$

Lens distortion model is often included to account for the distorting effect with the lens. The warping effect can be modeled using Brown-Conrady model in which a distorted pixel can be compactly expressed as

$$\bar{\mathbf{X}}_{\text{distort}} = (1 + k_1 r^2 + k_2 r^4) \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} + \begin{pmatrix} l_1 & l_2 \end{pmatrix} \begin{pmatrix} 2\bar{x}\bar{y} & r^2 + 2\bar{y}^2 \\ r^2 + 2\bar{x}^2 & 2\bar{x}\bar{y} \end{pmatrix} \quad (7.3)$$

where  $\begin{pmatrix} \bar{x} & \bar{y} \end{pmatrix}^T$  is the normalized image projection  $\mathbf{C} \begin{pmatrix} x/z & y/z \end{pmatrix}^T$ , and  $r$  represents norm  $\left( \begin{pmatrix} \bar{x} & \bar{y} \end{pmatrix}^T \right)$ . The radial and tangential distortion coefficients are denoted by  $k_{1-2}$  and  $l_{1-2}$ , respectively. In this lab, we will use a simplified model where the imaging plane is parallel to the work surface. Hence, relation between the image coordinates and world reference frame reduces to 1-dof rotation and 2-dof translation and scaling. Since the 2-dof scaling can be determined independently based on the parallel constraint, the camera calibration process reduces to obtaining the orientation and translation of the work surface and imaging plane.

#### 7.4.2 The Robot-Camera Calibration Process

In order to guide robot manipulation using images acquired from the camera, we need to first know the spatial relationship between the camera and the robot. Combining with the camera calibration matrix, we will be able to, under the appropriate specified constraints, map image coordinates to robotic joint coordinates for path planning and execution to accomplish manipulation tasks. There are many robot-camera calibration approaches for different conditions and application.

To simplify the demonstration of robot-camera calibration process, this lab will use an eye-to-hand model, where the camera is fixed with respect to the world coordinate system with known spatial relationship to the robot base. Combing the camera calibration model, we can now map image coordinates to world coordinates referenced from the base and subsequently the robotic joint coordinates.

### 7.5 Lab Activities

In this lab we will use the image processing techniques demonstrated in previous lab to detect, and label objects in the camera scene hence guiding manipulation tasks in object classification and sorting as illustrated below.

- Transform Image (Pixel) Coordinates to Robotic Reference Frame



Figure 7.2: Workflow for the task of image-guided robotic object sorting

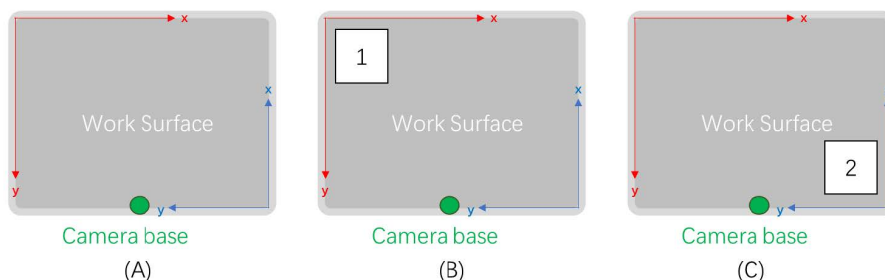


Figure 7.3: Registering two positions to find image-robot relationship

- Define relationship between the robot base frame and the pixel coordinate frame as illustrated by the red and blue axes, respectively.
- Move robot to Region 1: pixel coordinates  $(x_1, y_1)$ ; register the robot coordinates as  $(x'_1, y'_1)$ .
- Move robot to Region 2: pixel coordinates  $(x_2, y_2)$ ; register the robot coordinates as  $(x'_2, y'_2)$ .
- Obtain the ratio that map  $(x, y)$  to  $(x', y')$  as define below:

$$\begin{aligned} x_{ratio} &= (x'_2 - x'_1) / (y_2 - y_1) \\ y_{ratio} &= (y'_2 - y'_1) / (x_2 - x_1) \end{aligned} \quad (7.4)$$

$$\begin{aligned} x' &= x'_1 + x_{ratio} * (y - y_1) \\ y' &= y'_1 + y_{ratio} * (x - x_1) \end{aligned} \quad (7.5)$$

- Classify Objects and Compute Pose (Lab 5: Image Processing)
  - Label the objects and
  - Compute their center positions and orientation in the scene
- Move Robot using on Transformed Coordinates Move robot arm from one position to another as you have done in Lab 4 (inverse kinematics) with the Transformed Coordinates,
  - Locate objects and check their label as in the image
  - Transform coordinates to robot coordinate system
  - Move to pick (suction grip) object from the unsorted tray as should in Figure 7.5

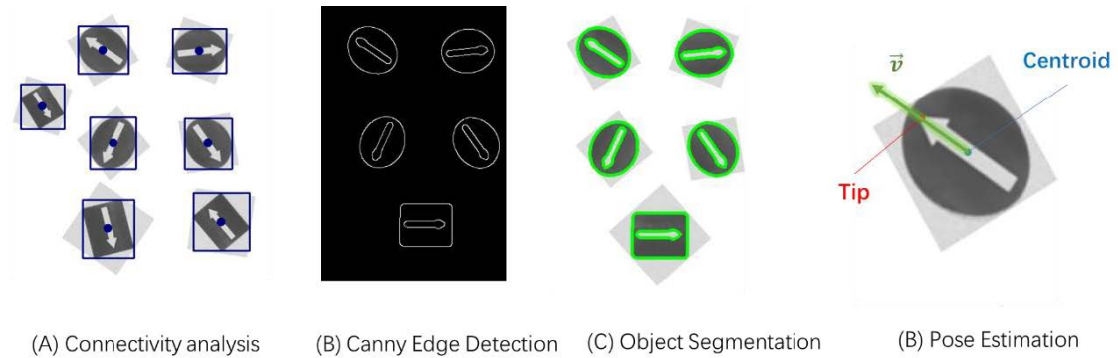


Figure 7.4: Image processing to classify and localize object in camera scene

- Move to sort objects into specific pose in the sorted tray as shown in Figure 7.5

## 7.6 Debugging

This lab can be a bit tricky to work through as several parts require trial and error. Here are some suggestions:

- Approach the lab sequentially and make sure you get good results at each step.
- Comment out parts that you are not currently using (Don't forget the uncomment them though...)

## 7.7 Report

This lab does not require a formal lab report - only the submission of your code. You should submit a document to [Blackboard](#) containing all of the files that you edited to complete the lab. This will be submitted to [Blackboard](#) as in the past.

## 7.8 Demo

Your TA will require you to run your program twice, each time with different initial block positions and verify that your program can locate the blocks and place them in the correct location. You will also demonstrate that suction feedback is working.

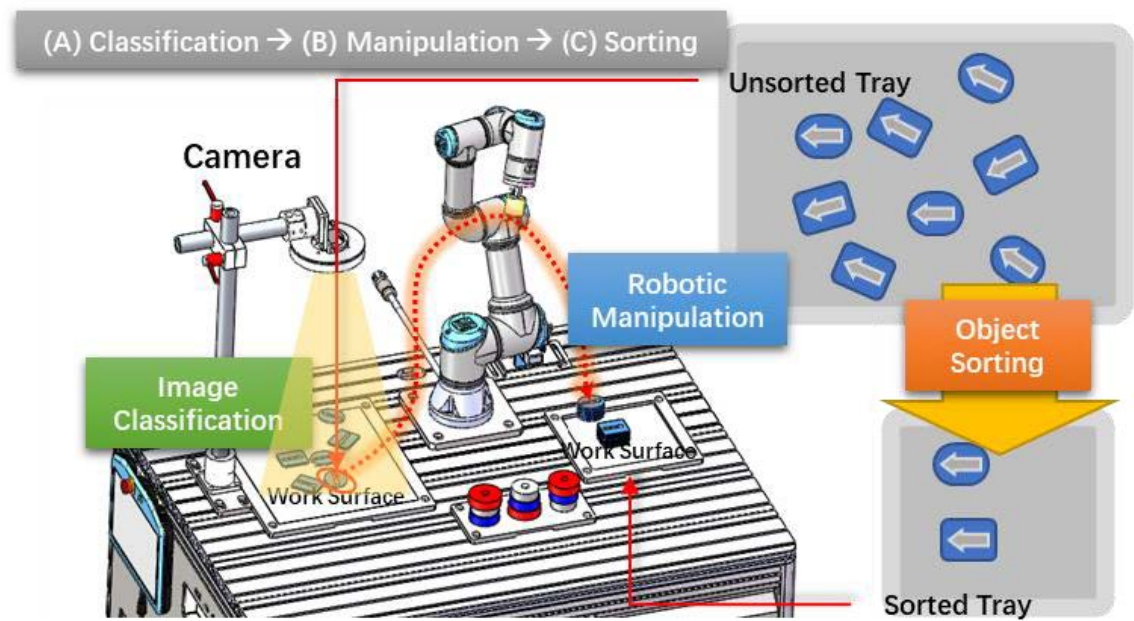


Figure 7.5: Image processing to classify and localize object in camera scene

## 7.9 Grading

- 100 points, successful demonstration and submission of the code. (No code = 0 points)

## Appendix B

# Notes on Camera Calibration and Transformations

### B.1 Simplified Perspective Transform

To extract spatial information from camera images, we need to derive equations to compute an object's coordinates in the world frame  $(x, y, z)_w$  based on row and column pixel coordinates in the image  $(r, c)_c$ . Instead of giving you these equations, this section will walk through how the transform works to allow you to derive them yourself.

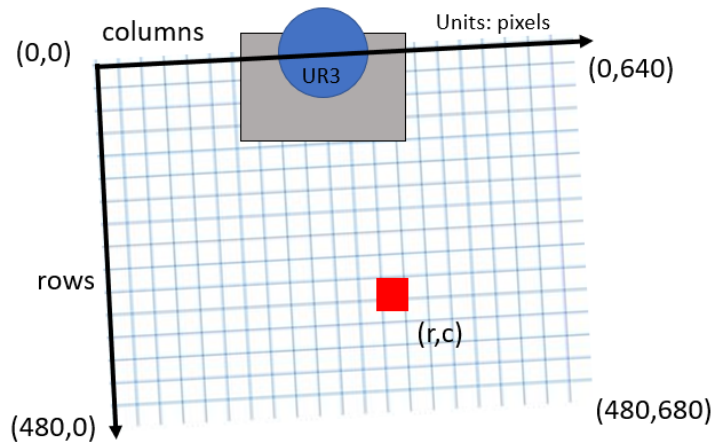


Figure B.1: The view from the camera.

The image starts as seen in Figure B.1. We can see that all data is in pixels, so

### B.1. SIMPLIFIED PERSPECTIVE TRANSFORM

the location of the object would be described by the rows and columns of the pixel at the centroid of the red block. We can further see that the image is not square to the table, but instead slightly rotated. We will ignore that for the time being.

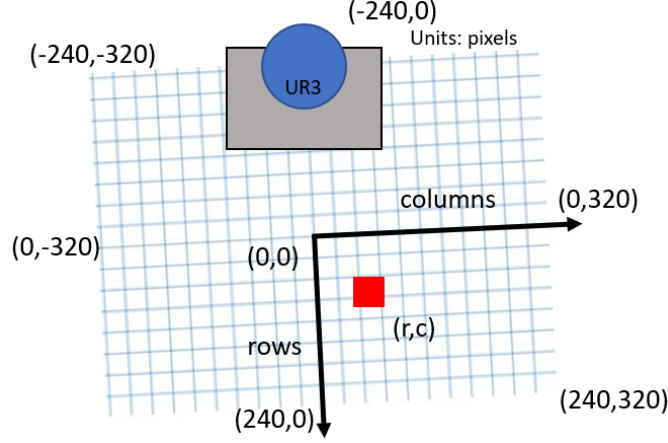


Figure B.2: The view from the camera with the origin shifted to the principal point.

The first step is to translate the frame from the top left corner, to the center of the image also known as the principal point as seen in Figure B.2. As seen in Table B.1, the location of the principal point in our original frame are called  $(O_r, O_c)$ . Based on this image, what are  $O_r$  and  $O_c$

$$\begin{aligned} O_r &= \\ O_c &= \end{aligned}$$

Side note: It is possible to develop the camera transformation without this translations, but shifting to the principle point is the standard method. It is recommended that you consult a more exhaustive text on computer vision to better understand why.

The next step is to move from pixel measurements to something we can measure in the real world such as meters. To do this, we need a relationship between pixel distance and physical distance on the table top. This value depends both on the physical properties of the camera and the distance between the camera and table top. The physical properties of the camera are encapsulated by focal length ( $f_x$  and  $f_y$  as seen in Table B.1). Because we do not know the focal length of the camera, we instead create a simple value  $\beta$  which encapsulates both pieces of information:

$$\beta = \frac{f_x}{z_c} = \frac{f_y}{z_c}.$$



### B.1. SIMPLIFIED PERSPECTIVE TRANSFORM

---

This is a simplification of this relationship. It assumes that pixels are square and that the lens does not distort the image at all. Because of this, you might find it is less accurate at the edges of the image, but it should work fine for our purposes. The units of  $\beta$  are pixels per meter.

We now have enough information to complete our first equation. This equation gives us the location of the block in meters, but with respect to the camera frame and not the world frame. These equations are often called the **intrinsic** equations because they rely on the intrinsic properties of the camera such as  $O_r$ ,  $O_c$ ,  $f_x$  and  $f_y$  (i.e. properties that don't change). These equations are essentially a transition from the camera sensor to the tabletop as seen in Figure B.3. Please complete the intrinsic equations using only  $\beta$ ,  $O_r$ , and  $O_c$

$$\begin{aligned} x_c(r) &= \\ y_c(c) &= \end{aligned}$$

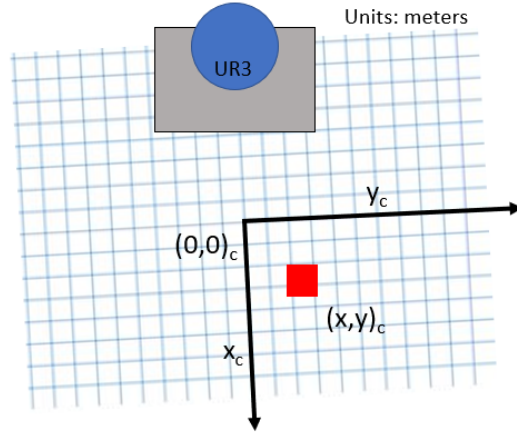


Figure B.3: The camera frame projected onto the table top. This is the output of the intrinsic equations.

Now that we have the intrinsic equations and have transitioned to the table top, we can try to align this table top camera frame with the world frame at the corner of the aluminum plate. We start this by shifting the frame from its current position which sits where the principal point lines up on the table top as shown in Figure B.4. As seen in Table B.1, the relationship between these two frames is  $O_{cw}$  which is the origin of the world frame in terms of the camera frame. The key values here are  $T_x$  and  $T_y$  as  $T_z = 0$ .

Now that we have aligned the origins, it is time to straighten up the frame as shown in Figure B.5. We do this naturally by a rotation about the z-axis by the angle  $\theta$ . We can see in Table B.1 that  $\theta$  is defined for  $R_{cw}$ .

Now that we have translated and rotated our frame to align with the world frame, we can write the **extrinsic** equations. These are known as extrinsic

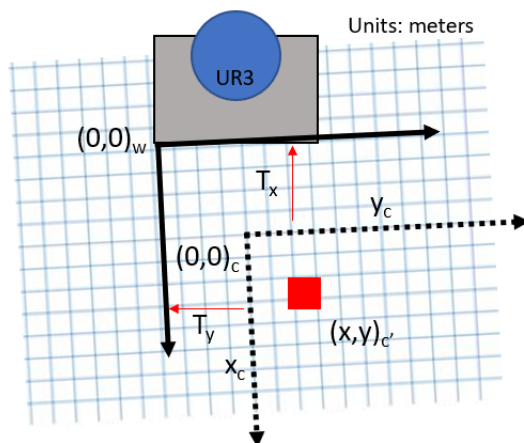


Figure B.4: The table top camera frame must be translated to align with the world frame.

equations because they are based on values outside of the camera and change depending on placement and orientation. Let's write the extrinsic equations:

$$\begin{bmatrix} x_w \\ y_w \end{bmatrix} =$$

Pay special attention to the signs of  $T_x$  and  $T_y$ !

We now have both the intrinsic and extrinsic equations and we can put them both together:

$$\begin{aligned} x_w(r, c) &= \\ y_w(r, c) &= \end{aligned}$$

These are the equations that you will use in your code to perform the camera transformation. What about  $z_w$ ? Well, since we are working on the flat table top,  $z_w$  is just the height of any object we wish to pick up. A value for  $z_w$  is provide in the code.

## B.2 Camera Calibration

After formulating the camera transformation it is natural to ask how the key values such as  $\beta$ ,  $\theta$ ,  $T_x$  and  $T_y$  are found. This is through the process of camera calibration. Due to simulation limitations, this has already been done for this lab. This section will give a brief introduction to how this process works.

- $\beta$  - This value is found using a calibration stick. The stick has two target set 10cm apart (center to center). When the centroids of these targets are

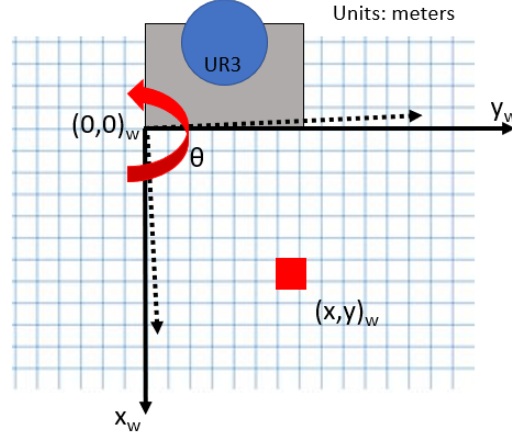


Figure B.5: The table top camera frame must be rotated to align with the world frame.

Name	Description
$(r, c)$	(row,column) coordinates of image pixel
$f_x, f_y$	ratio of focal length to physical width and length of a pixel
$(O_r, O_c)$	(row,column) coordinates of <i>principal point</i> of image (principal point of our image is center pixel)
$(x_c, y_c)$	(meters) coordinates of point in camera frame
$O_{cw}$	$= [T_x T_y T_z]^T$ origin of world frame expressed in camera frame
$R_{cw}$	rotation expressing world frame w.r.t. camera frame.

Table B.1: Definitions of variables necessary for camera calibration.

found, their distance in pixels can be determined. It is then just a matter of relating the pixels to the fixed 10cm distance.

- $\theta$  - The same calibration stick is used again. It is placed on the table top such that it carefully aligns with the y-axis of the world frame. When the centroids of the targets are found, a right triangle can be formed with the two points and the angle formed will be  $\theta$ .
- $T_x$  and  $T_y$  - This time a single target is placed on the table top with a known position in the world frame. With this known positions, the location of the centroid  $(r, c)$  and  $\beta$  and  $\theta$ , the only unknown in the camera transformation equations are  $T_x$  and  $T_y$ , which can easily be solved for.