Camera Calibration using AprilTag

(Assignment: 01)
Course: RME-5102

Name: Ahsan Imran

March 2, 2022

1 Experiment Goal

Implement a Project to Calibrate a Camera using Apriltag.

2 Theoretical Background

Geometric camera calibration, also referred to as camera resectioning, estimates the parameters of a lens and image sensor of an image or video camera. We can use these parameters to correct for lens distortion, measure the size of an object in world units, or determine the location of the camera in the scene. These tasks are used in applications such as machine vision to detect and measure objects. They are also used in robotics, for navigation systems, and 3-D scene reconstruction. To determine a 3D point's projection onto the image plane, at first we should transform the point from the world coordinate system to the camera coordinate system using extrinsic parameters (Rotation $\mathbf{R}$ and Translation $\mathbf{t}$). Next, using the intrinsic parameters of the camera, we project the point onto the image plane. The equations that relate $3D$ point (X_w, Y_w, Z_w) in world coordinates to its projection (u, v) in the image coordinates are shown below:

$$\begin{bmatrix} u' \\ v' \\ z' \end{bmatrix} = \mathbf{P} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix}$$

Where, $\mathbf{P}$ is a 3×4 Projection matrix consisting of two parts: the intrinsic matrix ($\mathbf{K}$) that contains the intrinsic parameters and the extrinsic matrix ($[\mathbf{R} \mid \mathbf{t}]$) that is combination of 3×3 rotation matrix $\mathbf{R}$ and a 3×1 translation $\mathbf{t}$ vector.
$\mathbf{P}$ = K x $[\mathbf{R} \mid \mathbf{t}]$ Here, the intrinsic matrix $\mathbf{K}$:

$$\mathbf{K} = \begin{bmatrix} f_x & \gamma & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix}$$

where, f_x, f_y are the x and y focal lengths o_x, o_y are the x and y coordinates of the optical center in the image plane. Using the image's center is usually a good estimate. γ is the skew between the axes.

3 Camera Specification

In this experiment, I have used the Redmi Note 4 mobile's back camera which has 13MP resolution. It can capture 1080 x 1920 pixels image.

4 Tag specification

Figure 1 is showing the Apriltags used in this experiment. Figure 2 showing examples of four images from total eight images that I have used to calibrate the camera. More details given below:
tagArrangement = [5,8];
tagFamily = 'tag36h11';
tagids=[1,2,3,...,40];

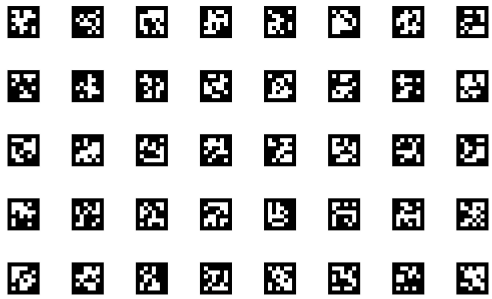

Figure 1: Apriltags used in this experiment

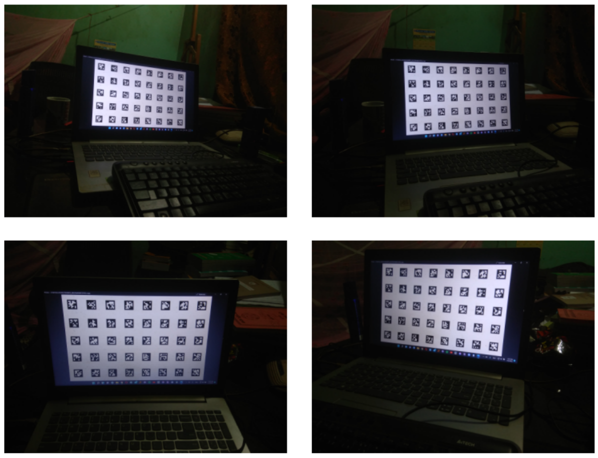

Figure 2: Captured Images

5 Procedure

Camera calibration procedure is as follows:

1. At first, We generated the calibration pattern by downloading and preparing pre-generated tags for all the supported families using Matlab 2021b.

2. We implement the matlab function called *helperDetectAprilTagCorners* that looks for a tag board and returns the coordinates of the corners (See Fig 3). Its usage is given by *[imagePoints, boardSize] = helperDetectAprilTagCorners(imdsCalib, tagArrangement, tagFamily)* that detect, and localize the tags from the captured images and arrange them in a checkerboard fashion to be used as key points in the calibration procedure. Here, imdsCalib is the image taken by our camera, tagArrangement is the row nd coloum number of the tag (row = 5, coloum = 8), and tagFamily indicate the tag name (we use tag tag36h11). Matlab's function *helperDetectAprilTagCorners* takes in the original image, and the location of corners, and looks for the best corner location inside a small neighborhood of the original location. The algorithm is iterative in nature and therefore we need to specify the termination criteria (e.g. number of iterations and/or the accuracy).

3. The final step of calibration is to pass the 3D points in world coordinates and their 2D locations in all

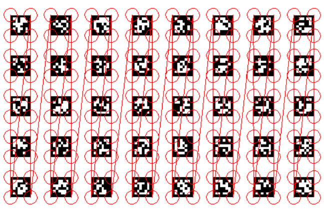

Figure 3: Corners

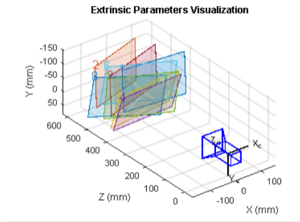

Figure 4: Extrinsic Parameter Visualization

images to Matlab's calibrateCamera method. The implementation is based on a paper by Zhengyou Zhang [Zha00]. The syntax of the code is *params = estimateCameraParameters(imagePoints, worldPoints)*, that returns cameraParams, a cameraParameters object containing estimates for the intrinsic and extrinsic parameters and the distortion coefficients of a single camera. Here, worldPoints is the 3D location of the corners with respect to world view coordinate and imagepoints are 2D locations of the corners with respect to image frame. The function also returns the images we used to estimate the camera parameters and the standard estimation errors for the single camera calibration. The estimateCameraParameters function estimates extrinsics and intrinsics parameters. Figure is the visualization of extrinsic parameter.

6 Result & Discussion

parameter	value
focal length	$\begin{pmatrix} 3.1969e+03 \\ 3.1970e+03 \end{pmatrix}$
principle point	$\begin{pmatrix} 2.0662e+03 \\ 1.5329e+03 \end{pmatrix}$
radial distortion	$\begin{pmatrix} 0.1286 \\ -0.1891 \end{pmatrix}$
tangential distortion	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$
Skew	0

Table 1: Intrinsic Parameters

Image	rotation	translation	Image	rotation	translation
Image 1	$\begin{bmatrix} 0.9983 & 0.0459 & -0.0357 \\ -0.0501 & 0.9906 & -0.1269 \\ 0.0295 & 0.1285 & 0.9913 \end{bmatrix}$	$\begin{pmatrix} -80.1609 \\ -130.3248 \\ 440.5438 \end{pmatrix}$	Image 2	$\begin{bmatrix} 0.9306 & 0.0613 & 0.3609 \\ -0.0298 & 0.9953 & -0.0923 \\ -0.3648 & 0.0752 & 0.9280 \end{bmatrix}$	$\begin{pmatrix} -113.3904 \\ -112.0783 \\ 535.9280 \end{pmatrix}$
Image 3	$\begin{bmatrix} 0.9606 & -0.0304 & 0.2763 \\ 0.0356 & 0.9993 & -0.0138 \\ -0.2756 & 0.0230 & 0.9610 \end{bmatrix}$	$\begin{pmatrix} -97.5395 \\ -95.2518 \\ 480.8397 \end{pmatrix}$	Image 4	$\begin{bmatrix} 0.9989 & -0.0005 & 0.0463 \\ -0.0137 & 0.9516 & 0.3070 \\ -0.0442 & -0.3073 & 0.9506 \end{bmatrix}$	$\begin{pmatrix} -131.0998 \\ -76.1571 \\ 372.0039 \end{pmatrix}$
Image 5	$\begin{bmatrix} 0.9974 & -0.0419 & 0.0583 \\ 0.0197 & 0.9405 & 0.3391 \\ -0.0690 & -0.3371 & 0.9389 \end{bmatrix}$	$\begin{pmatrix} -123.5833 \\ -64.5307 \\ 396.9402 \end{pmatrix}$	Image 6	$\begin{bmatrix} 0.9546 & 0.1097 & -0.2769 \\ -0.0751 & 0.9883 & 0.1328 \\ 0.2882 & -0.1060 & 0.9517 \end{bmatrix}$	$\begin{pmatrix} -134.7527 \\ -156.7881 \\ 450.2932 \end{pmatrix}$
Image 7	$\begin{bmatrix} 0.9561 & 0.0917 & -0.2783 \\ -0.0881 & 0.9958 & 0.0254 \\ 0.2795 & 0.0002 & 0.9602 \end{bmatrix}$	$\begin{pmatrix} -156.8667 \\ -136.7361 \\ 415.3707 \end{pmatrix}$	Image 8	$\begin{bmatrix} 0.9306 & 0.1325 & -0.3412 \\ -0.1144 & 0.9908 & 0.0728 \\ 0.3477 & -0.0287 & 0.9372 \end{bmatrix}$	$\begin{pmatrix} -142.6464 \\ -96.7056 \\ 502.8383 \end{pmatrix}$

Table 2: Extrinsic Parameters

We can see in the Table 1 there are five Intrinsic parameters. These parameters are independent of Image frames. Focal length in x and y directions are not exactly same. Also, there are radial distortion in both axes. There are no tangential distortion. Skew is also zero.

In Table 2, we can see the rotation and translation matrix for every image. In my experiment, I have used eight images. Rotation matrix is 3x3 and translation matrix is 3x1 for each image. These parameters changes when image frames are different for a single camera.

References

[Zha00] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000.