ECE 8410: Computer Vision

Laboratory 2 Report: Camera Calibration and Pose

March 3, 2021

Mark Belbin (201618477)
Raylene Mitchell (201415247)
Rodney Winsor (200433886)

Part 1: Camera Calibration

For this lab, images were taken from a Raspberry Pi camera stream instead of a laptop webcam. For one of our group members' projects, camera calibration of the Pi camera is actually required, so this is a good exercise. About 20 images were taken and imported into MATLABs camera calibration app. The calibration result in the app is shown in Figure 1.

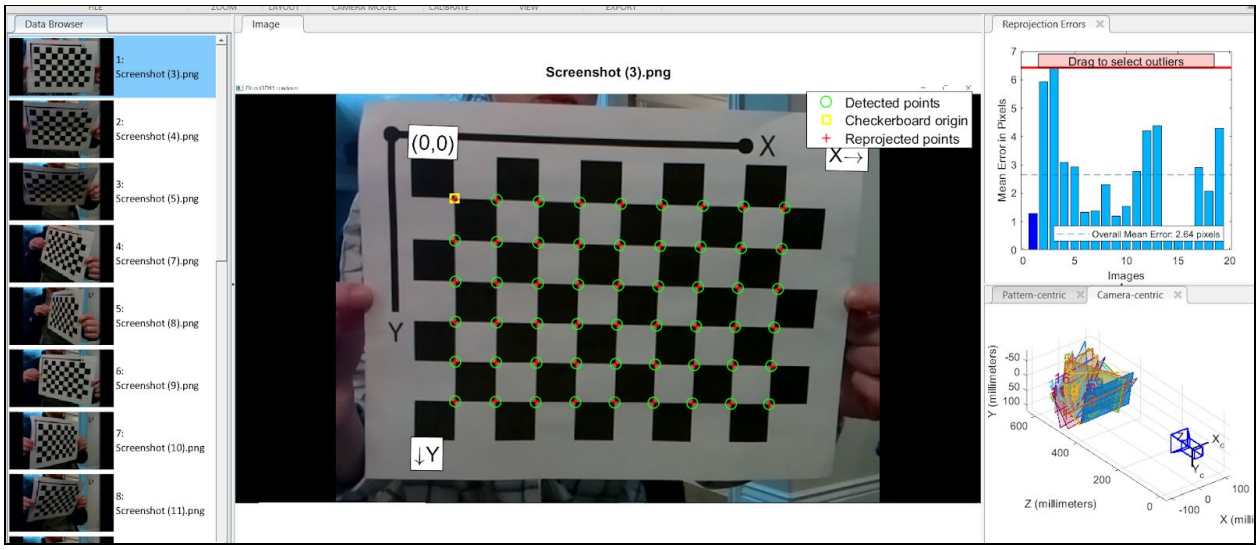

Figure 1: Camera calibration in MATLABs calibration app.

The following values recorded from the calibration are displayed in Table 1. The calibration may have been more accurate if the printed sheet was rigid, or taped to a rigid surface.

Table 1: Calibration data.

Focal Length X, fx [Pixels]	*1842.9*
Focal Length Y, fy [Pixels]	*1844.6*
Optical Center X, Cx [Pixels]	*952.8280*
Optical Center X, Cy [Pixels]	*488.9989*
Skew	*0*
First Image X Translation, tx [Pixels]	*-83.44*
First Image Y Translation, ty [Pixels]	*-40.76*
First Image Z Translation, tz [Pixels]	*400.30*
First Image X Rotation [Radians]	*0.1620*
First Image Y Rotation [Radians]	*-0.0678*
First Image Z Rotation [Radians]	*0.0275*

Part 2: Calculating Camera Pose Using Newton's Method

Using the provided MATLAB 'fproject' function and 'pose3D' script, the camera's pose relative to the defined coordinate system was calculated. The final pose for image one is displayed in Table 2, with a normalized pixel error of 3.49. It only took the algorithm 8 iterations to converge on this solution. The pixel locations of the given coordinates are plotted onto the original image 1 in Figure 2. As well, our group plotted the original defined coordinate pixel locations alongside the recalculated locations in Figure 3, which gives an idea of how good the final solution is.

Table 2: Final Camera Pose for Image 1.

Camera X Translation, tx [Units]	*8.4533*
Camera Y Translation, ty [Units]	*17.4056*
Camera Z Translation, tz [Units]	*59.0472*
Camera X Rotation [Radians]	*1.4012*
Camera Y Rotation [Radians]	*-0.5715*
Camera Z Rotation [Radians]	*0.1061*

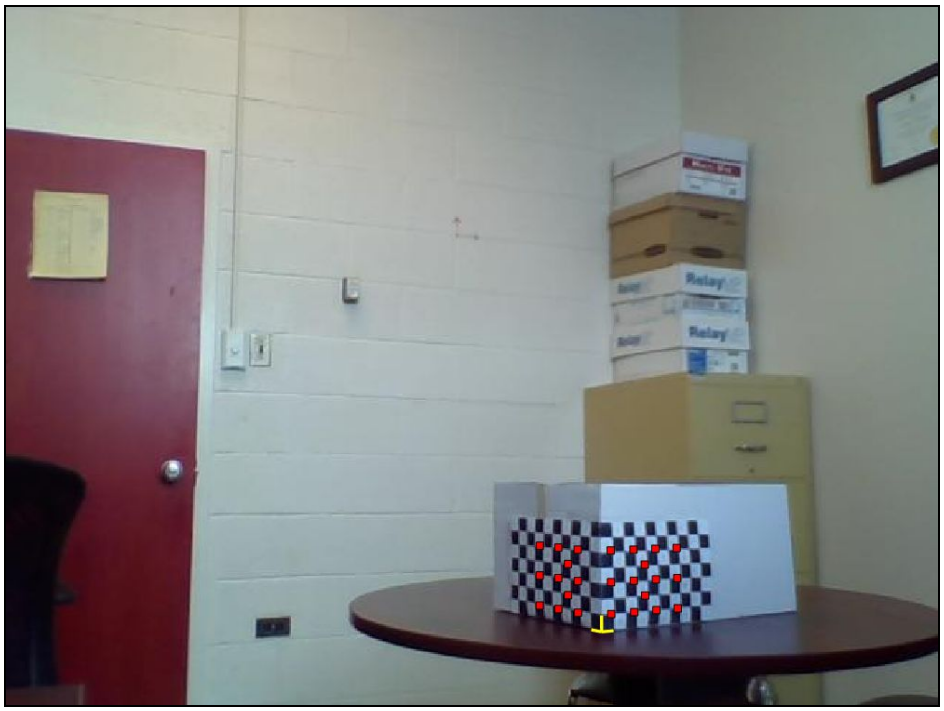

Figure 2: Final pixel locations of coordinates after pose iteration for image 1.

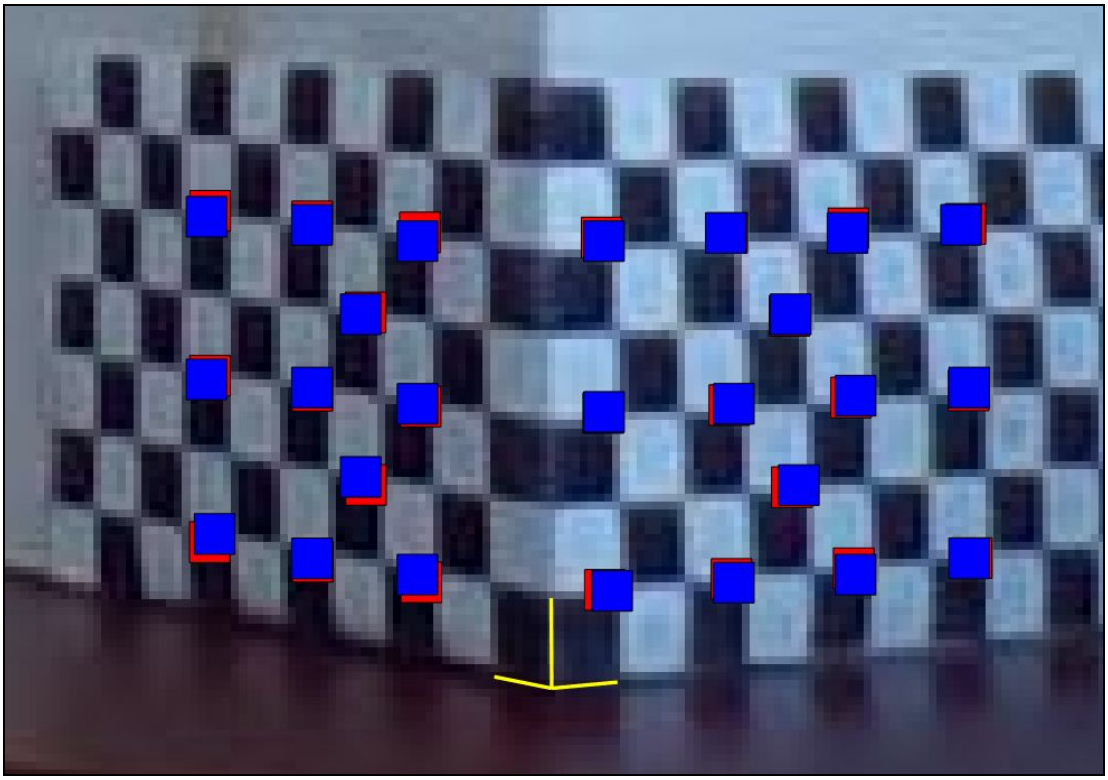

Figure 3: Calculated coordinate locations (red) and original specified coordinate locations (blue). The coordinate pixel locations that were calculated match near exactly to the defined locations.

As well, the camera pose was estimated for image 2. The calculated camera pose is displayed in Table 3, where the solution converged after 8 iterations with a normalized pixel error of 4.208. The resulting image with recalculated coordinate pixel locations is shown in Figure 4.

Table 3: Final Camera Pose for Image 2.

Camera X Translation, tx [Units]	*-16.328*
Camera Y Translation, ty [Units]	*16.8507*
Camera Z Translation, tz [Units]	*51.8341*
Camera X Rotation [Radians]	*1.2024*
Camera Y Rotation [Radians]	*-1.1887*
Camera Z Rotation [Radians]	*0.3397*

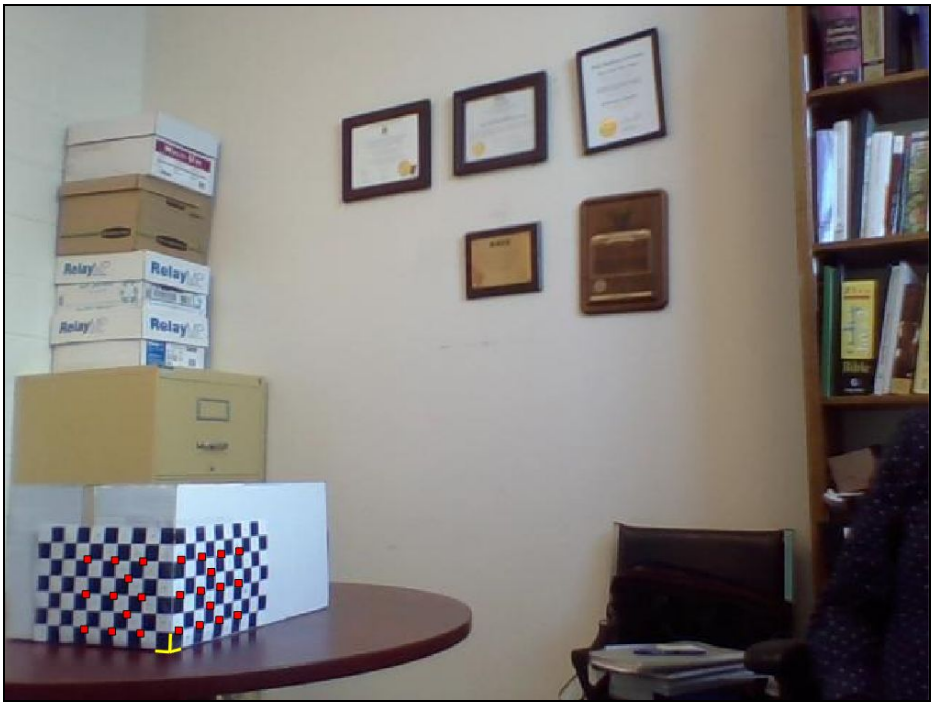

Figure 4: Final pixel locations of coordinates after pose iteration for image 2.