Daniel Perkins

Dr. Lee

EC EN 631

27 February 2025

Task 1

The positions of the four corners with respect to the left camera are:

$$\begin{bmatrix} -19.30 \\ -31.13 \\ 408.32 \end{bmatrix}, \begin{bmatrix} 15.56 \\ -29.89 \\ 409.09 \end{bmatrix}, \begin{bmatrix} -20.11 \\ -7.96 \\ 409.20 \end{bmatrix}, \text{and} \begin{bmatrix} 14.67 \\ -6.83 \\ 412.08 \end{bmatrix}$$

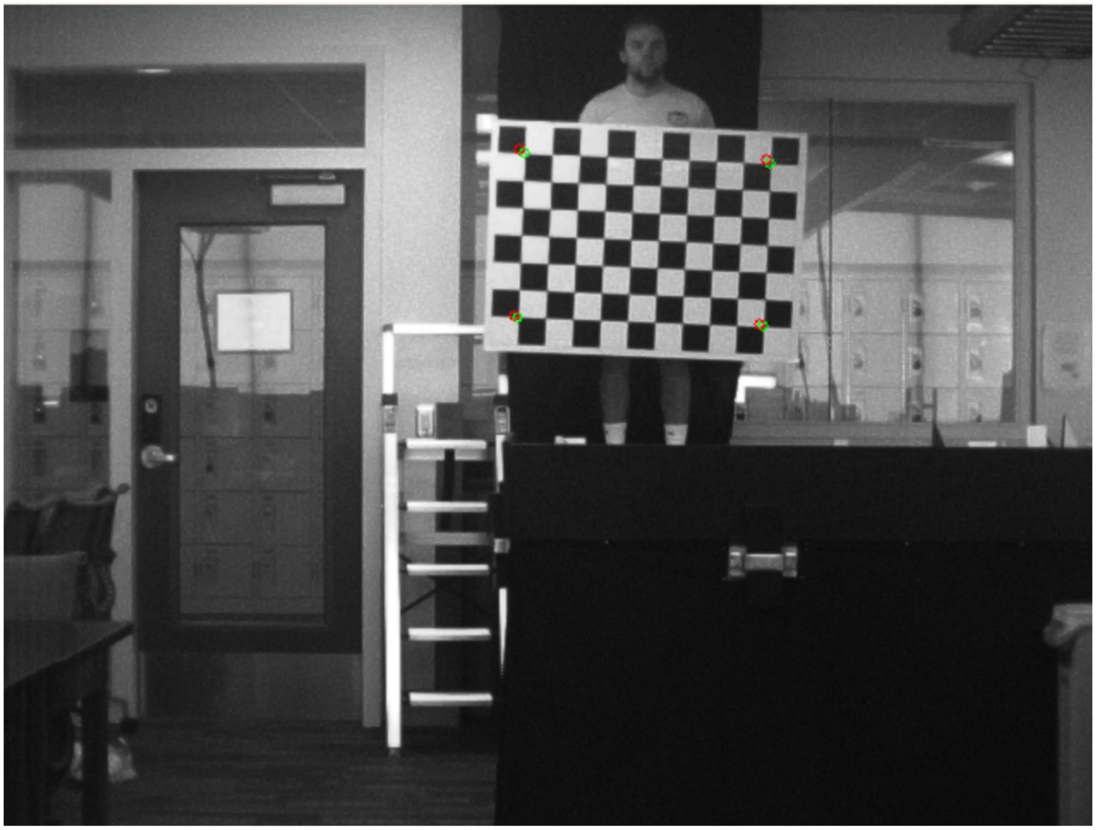
The positions of the four corners with respect to the right camera are:

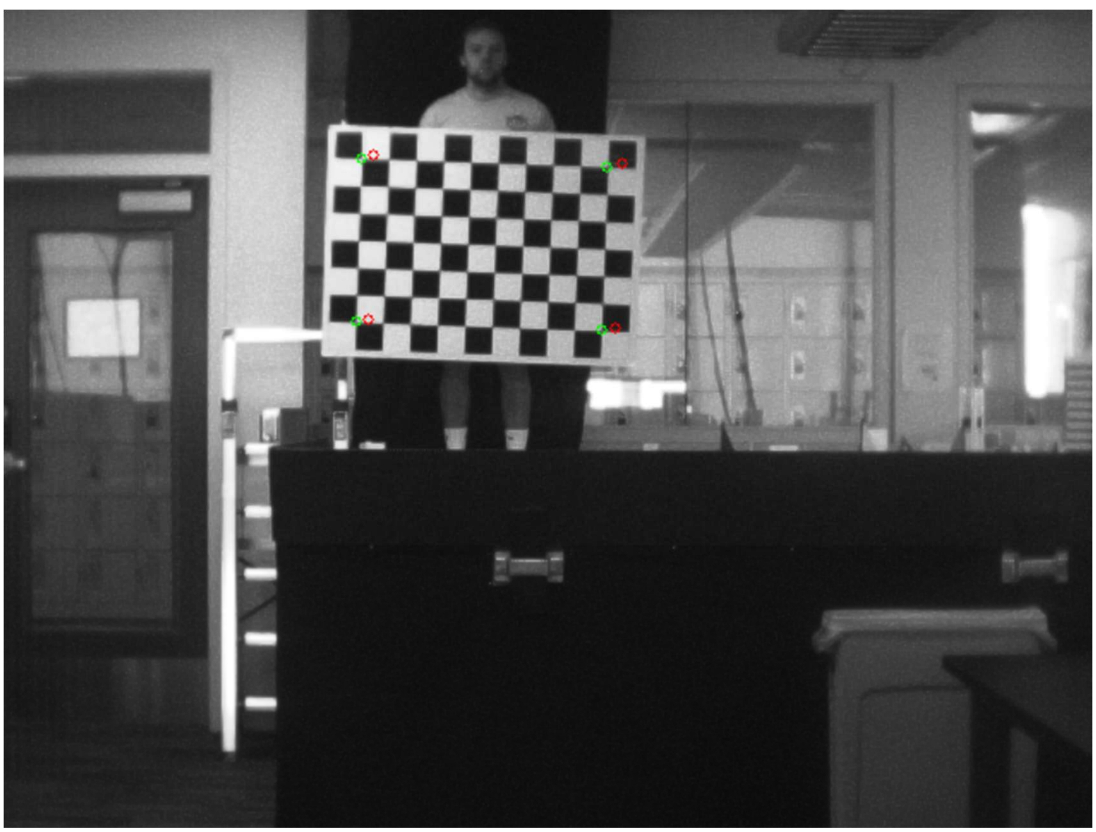
$$\begin{bmatrix} -39.69 \\ -31.16 \\ 408.93 \end{bmatrix}, \begin{bmatrix} -4.83 \\ -29.92 \\ 409.70 \end{bmatrix}, \begin{bmatrix} -40.50 \\ -7.99 \\ 409.81 \end{bmatrix}, \text{and} \begin{bmatrix} -5.72 \\ -6.86 \\ 412.70 \end{bmatrix}$$

These points are equal to the points from the left camera times the rotation matrix, plus the transition vector (P=RP+T) where:

$$\mathrm{R}=\begin{bmatrix} 0.99998 & 0.00320 & -0.00453 \\ -0.00313 & 0.99986 & 0.01656 \\ 0.00458 & -0.01654 & 0.99985 \end{bmatrix} \; and \;\; \mathrm{T} = \begin{bmatrix} -20.38998 \\ -0.029863 \\ 0.614177 \end{bmatrix}$$

The calculation/proof of this is in the code. Note that since this was calculated form undistorted and rectified points, we actually don’t need the rotation matrix.





Task 2

Now, using perspectiveTransform(), we got the following results:

The positions of the four corners with respect to the left camera are:

$$\begin{bmatrix} -19.30 \\ -31.15 \\ 408.32 \end{bmatrix}, \begin{bmatrix} 15.56 \\ -29.89 \\ 409.09 \end{bmatrix}, \begin{bmatrix} -20.11 \\ -7.95 \\ 409.20 \end{bmatrix}, \text{ and} \begin{bmatrix} 14.67 \\ -6.83 \\ 412.08 \end{bmatrix}$$

The positions of the four corners with respect to the right camera are:

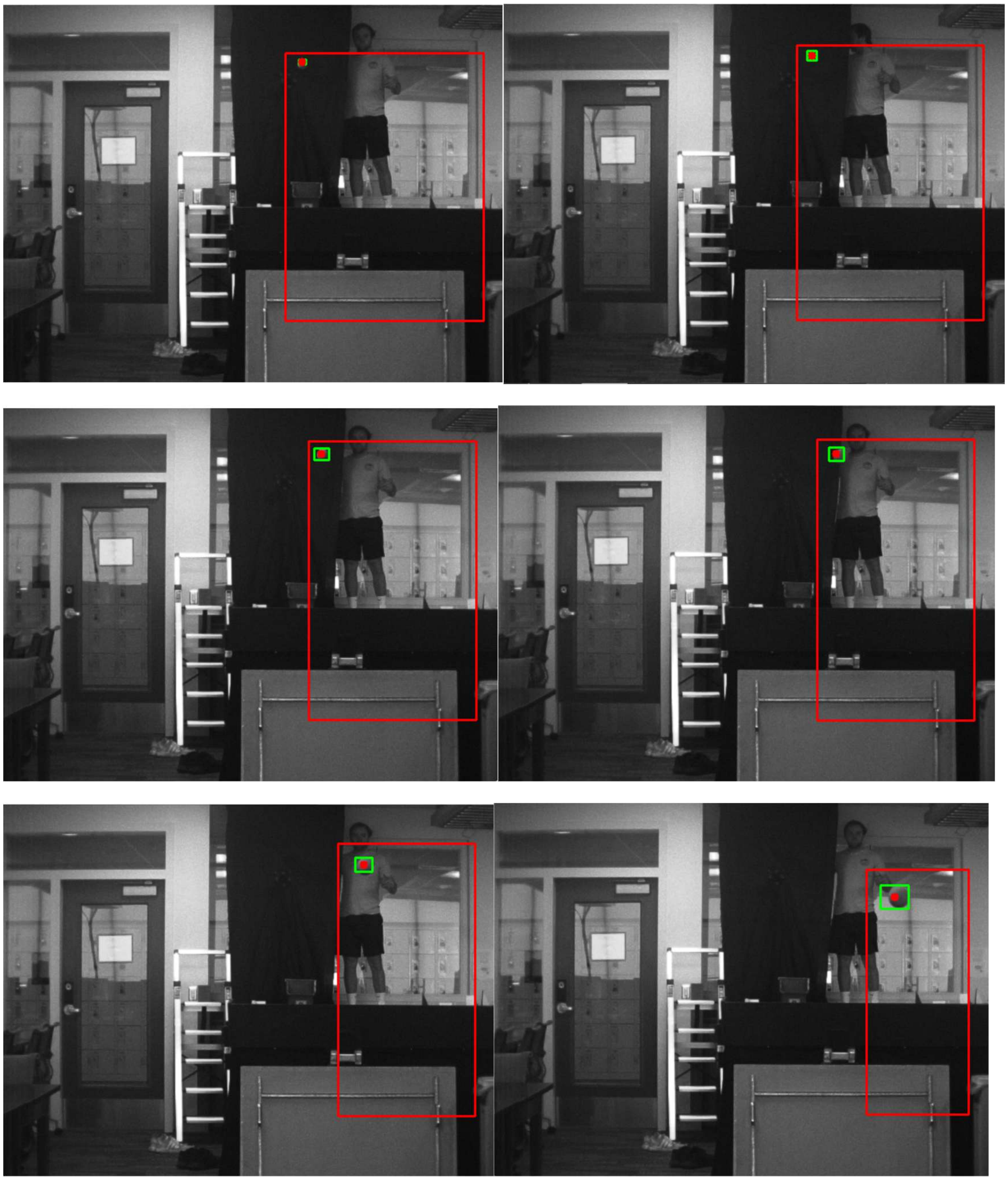
$$\begin{bmatrix} -39.70 \\ -31.12 \\ 408.32 \end{bmatrix}, \begin{bmatrix} -4.83 \\ -29.90 \\ 409.09 \end{bmatrix}, \begin{bmatrix} -40.51 \\ -7.97 \\ 409.20 \end{bmatrix}, \text{ and} \begin{bmatrix} -5.73 \\ -6.84 \\ 412.08 \end{bmatrix}$$

These results are essentially the same as the last task. They only differ a bit because of rectification. But, as we used the undistortPoints() function for both (which rectifies and undistorts the points), the difference is practically 0.

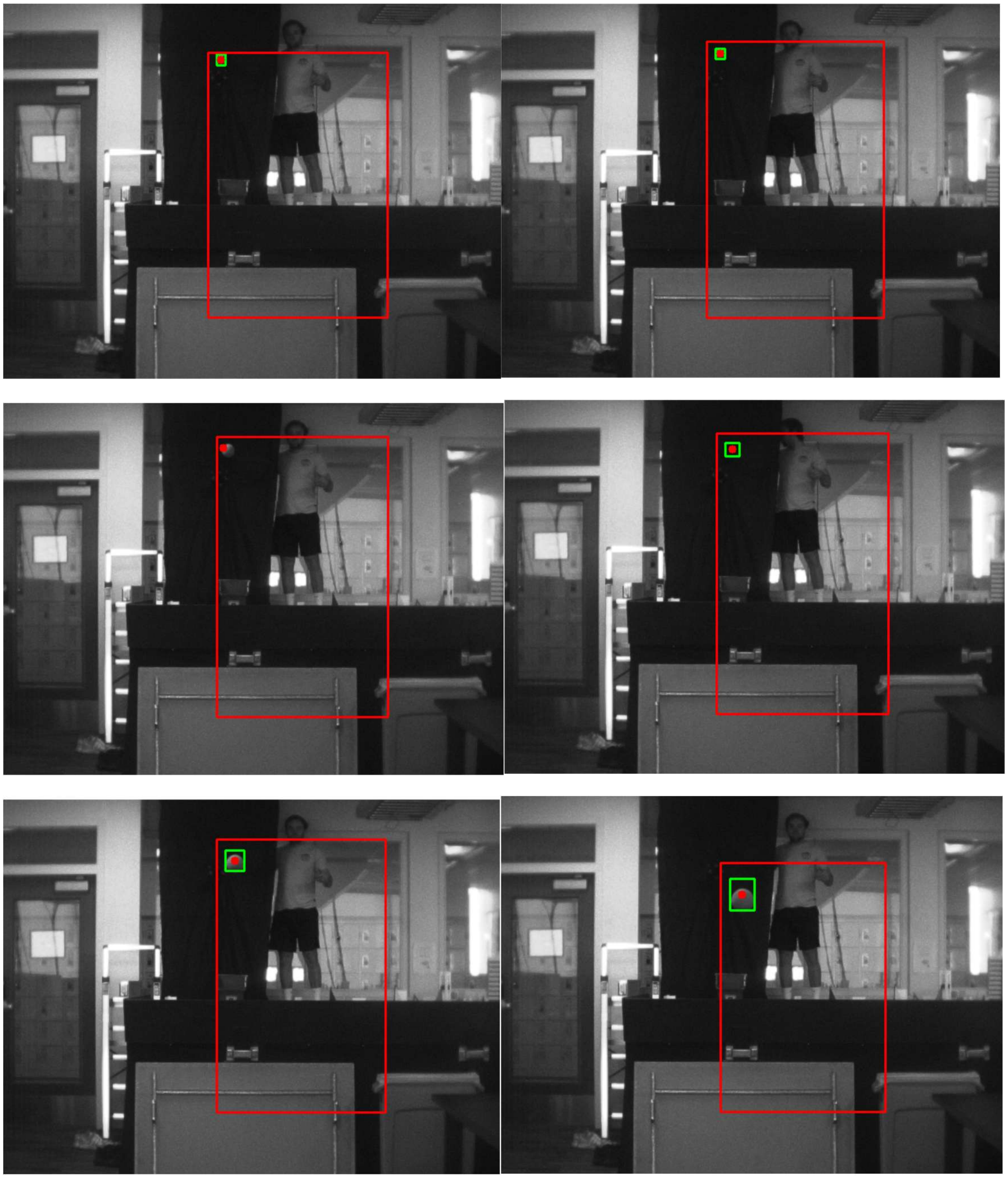
Also, as shown in the code, we have the relation P_l=P_r+[||T||, 0 ,0]^T. Furthermore, the distance between the top left point and bottom left point is approximately 6*3.88” as desired.

Task 3

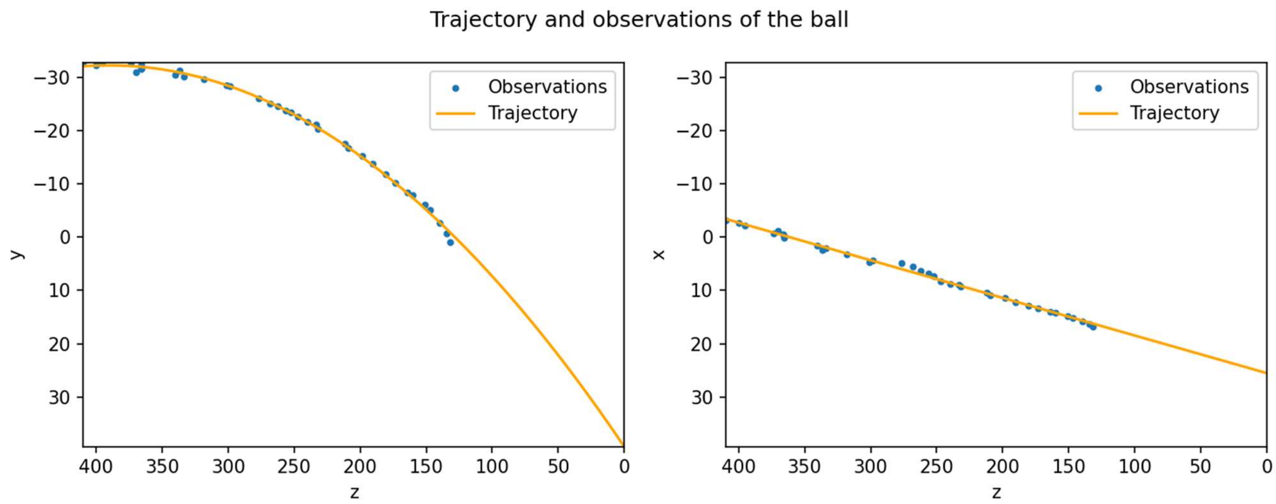
Left Camera



Right Camera



Task 4



I followed a similar process of that in task 1 to get the 3-D points of the baseball from the video. Then, to estimate the trajectory, I used polynomial interpolation. Rather than forcing it to be of high degree, for the (z,y) coordinates I used a 2nd degree to match the acceleration of gravity. And, for the (z,x) coordinates, I used a linear interpolation (as there should be no acceleration). This prevented the noise from harming our predictions.

Here are the final predictions for where the ball will end up:

Final x: 27.1967938897244

Final y: 39.30283408622389

Those are technically with respect to the left camera. If we add T/2, we can get the predictions with respect to the center of the cameras:

Final x: 15.341366805593035

Final y: 39.287902426185056