Brandon Botzer

AI 879 Week 13

Authors Note:

I had to have a kidney stone removed this past week and didn't have the time to work this out in python using OpenCV. The extra work needed in OpenCV is the undistorting of the images due to the affine transforms, calculating the disparity map, and then creating a depth map. Given that Matlab provides a nice GUI for calibration and bundles the depth map calculation into the triangulate() function, I present this work using Matlab and the StereoCameraCalibration App.

1. Camera Calibration

To run a triangulation method on stereo images, we first must know the stereo calibration parameters. These parameters are dependent on the physical properties of a camera. To obtain these parameters, I captured 12 images of a chessboard pattern at a 12-inch distance. The 12 images were taken at different angles to the chessboard pattern. The general setup can be seen in Figure 1.

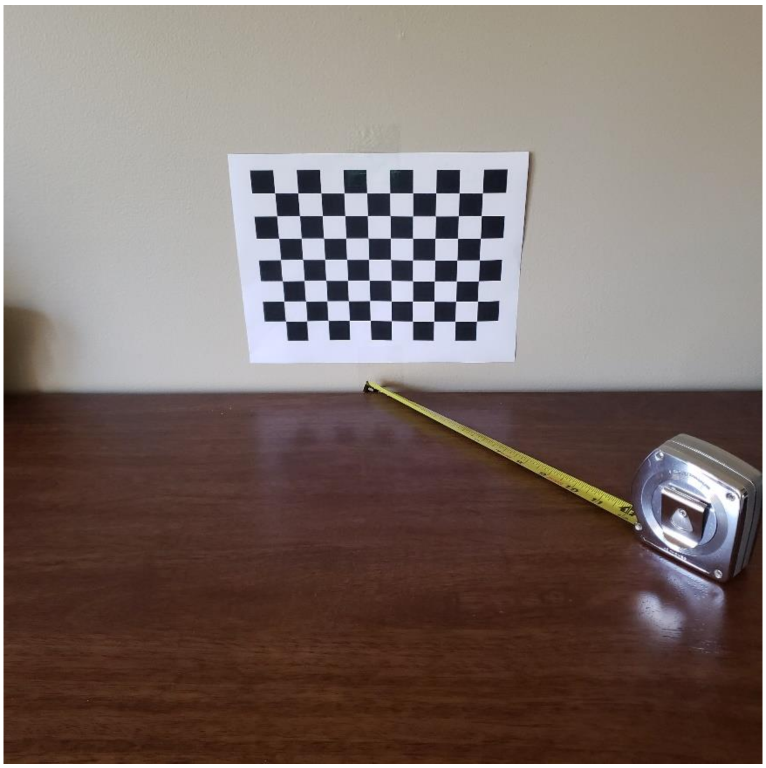

Figure 1: A chessboard pattern taped to the wall. The camera was placed at the 12-inch mark as given by the rule (bottom right). The author did his best to hold the camera at the same height each time by using the rule as a height guide.

These 12 images were taken and loaded into the Matlab StereoCameraCalibration App. Matlab was able to perform feature matching on the chessboard image as seen in Figure 2.

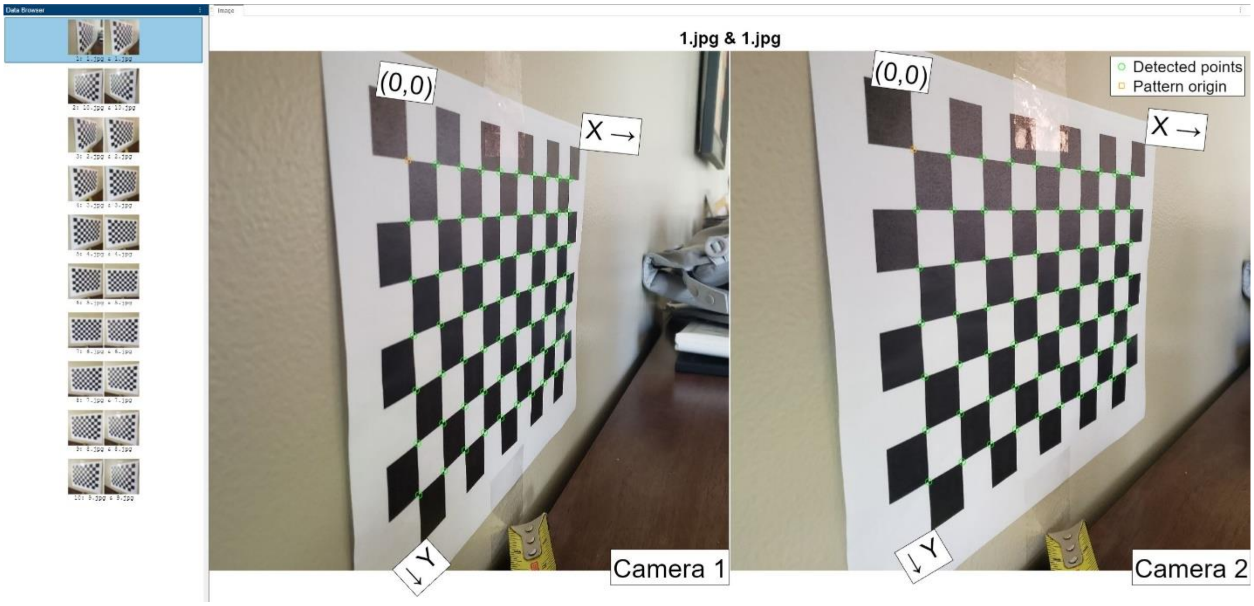

Figure 2: Corners were found on two 'adjacent' images.

From here, the parameters were calibrated. During calibration, there was one set of adjacent images that were rejected due to large enough differences that Matlab was unable to resolve the calibration. The real world printed square size was also provided as 0.857 inches.

As seen in Figures 3 and 4, the reprojection of the points via the calibration can vary from the truth.

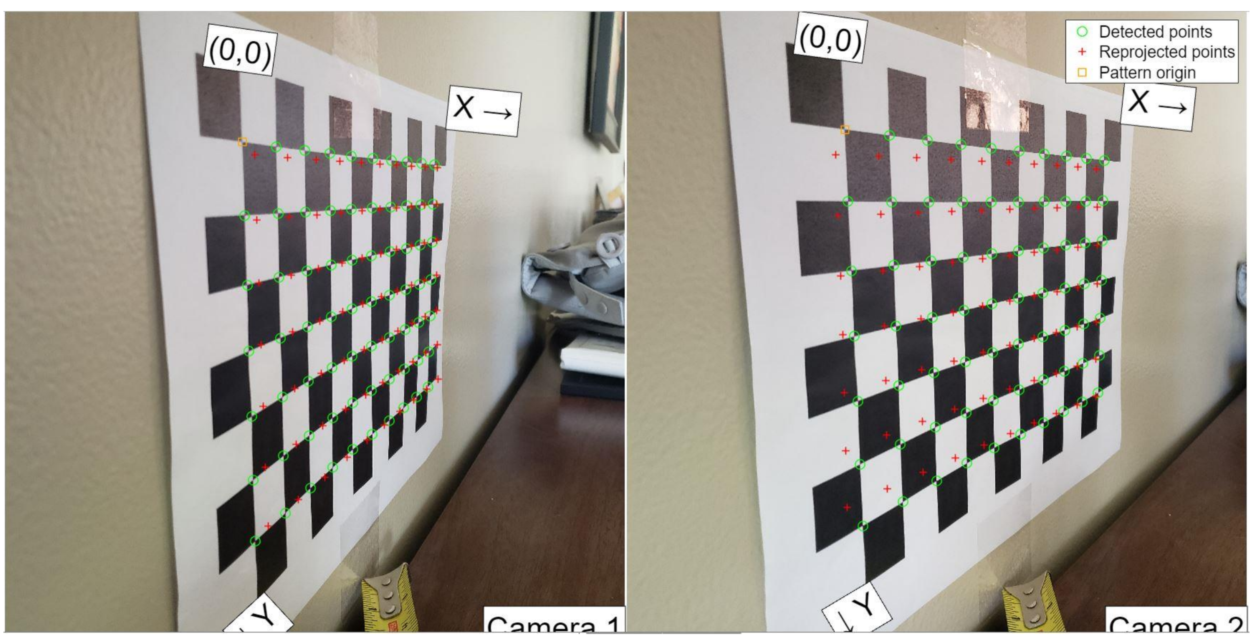

Figure 3: A poor reprojection of the image points from the calibration for images 1 and 2.

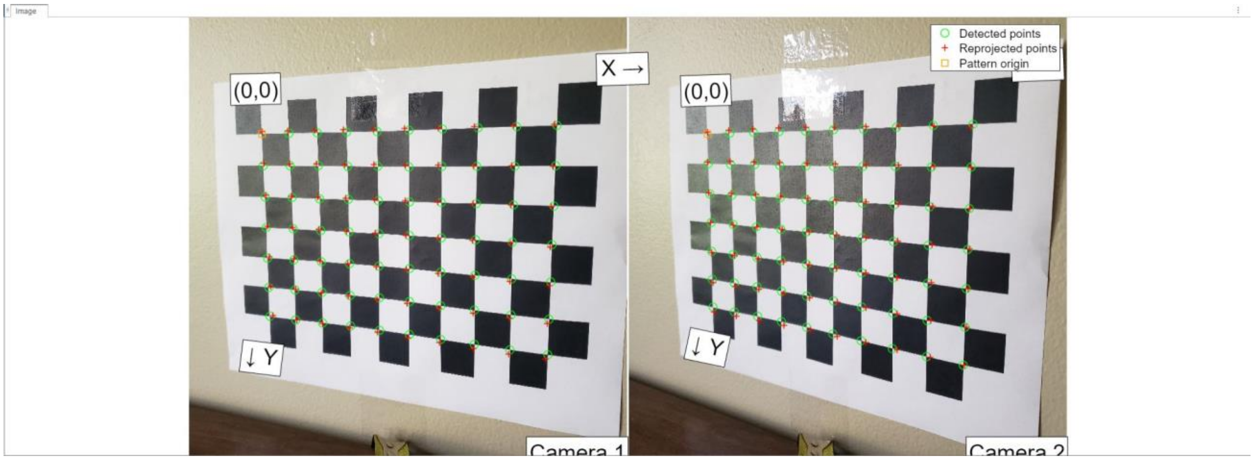

Figure 4: A better reprojection of the image points from the calibration for images 8 and 9.

Figure 3 shows a poor reprojection on the first two images taken while Figure 4 shows a better (though not perfect) reprojection on images 8 and 9.

In an effort to attempt to remove poor reprojections and obtain the best stereo parameters (stereoParams) outliers were selected and removed from the calibration dataset as show in Figure 5.

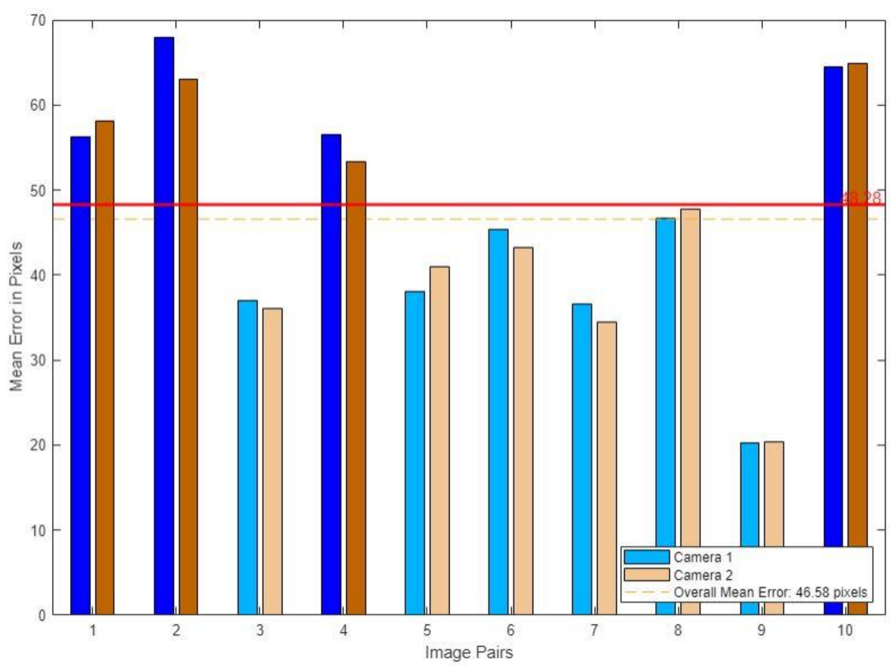

Figure 5:The error bar was lowered to remove data points 1, 2, 4, and 10 as their reprojection errors were high relative to the other image sets.

A view of how Matlab interpreted the camera positions can be seen in Figure 6. In the figure, you can see that it appears that there were more images taken from the left side of the chessboard (when looking at the chessboard these would be located to the right of the board) and that there is some perceived change in distance to the chessboard image despite my best attempts to take every photo at 12-inches.

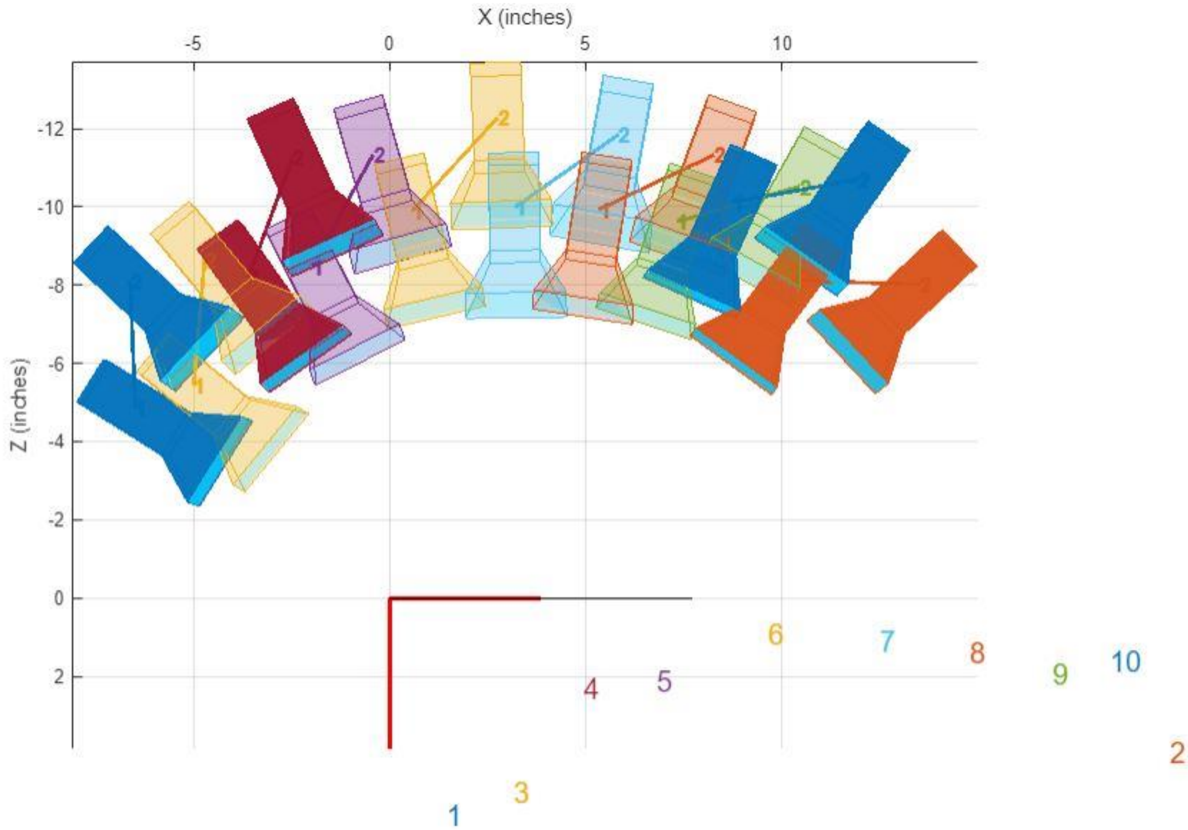

Figure 6:Relative camera positions for mapped images to the chessboard. This is an overhead display with the chessboard image at the bottom in red.

As the final step, the radial distortion, skew, and tangential distortions were computed and the parameters were exported.

2. Triangulation

With the stereoParams, triangulation of stereo images can be done to map the distance to the object. To do this, grayscale images where read in, features were mapped with the SURF algorithm, inliers were found using the RANSAC method, and then distances were found using Matlab's triangulation method.

A detailed look at the code can be seen in the follow-on pdf document.

3. Results

Firstly, there is next to zero documentation that I could find that would explain what units my depths were calculated in. I assume that the steroParams values carry over the inches that I prescribed but I have no way of confirming this other than looking at the values obtained and agreeing that they are most likely in units of inches.

As for the outputs from the triangulation() itself, Figures 7 and 8 show the output distances. In Figure 7, the mapped points were obtained without the use of the RANSAC method. As a result, while some values are near the 12-inches of distance the image was taken at, some of the reported values are in the hundreds. Figure 8 seeks to rectify this by applying the RANSAC method. In doing so, the most extream values in the hundreds were taken away and with many values being around the 12-inches. However, there are still some spurious values such as the 57.0 inches (units) still remain.

Figure 7: For images 8 and 9, triangulated distances to the chessboard image. This was done without the application of the RANSAC method.

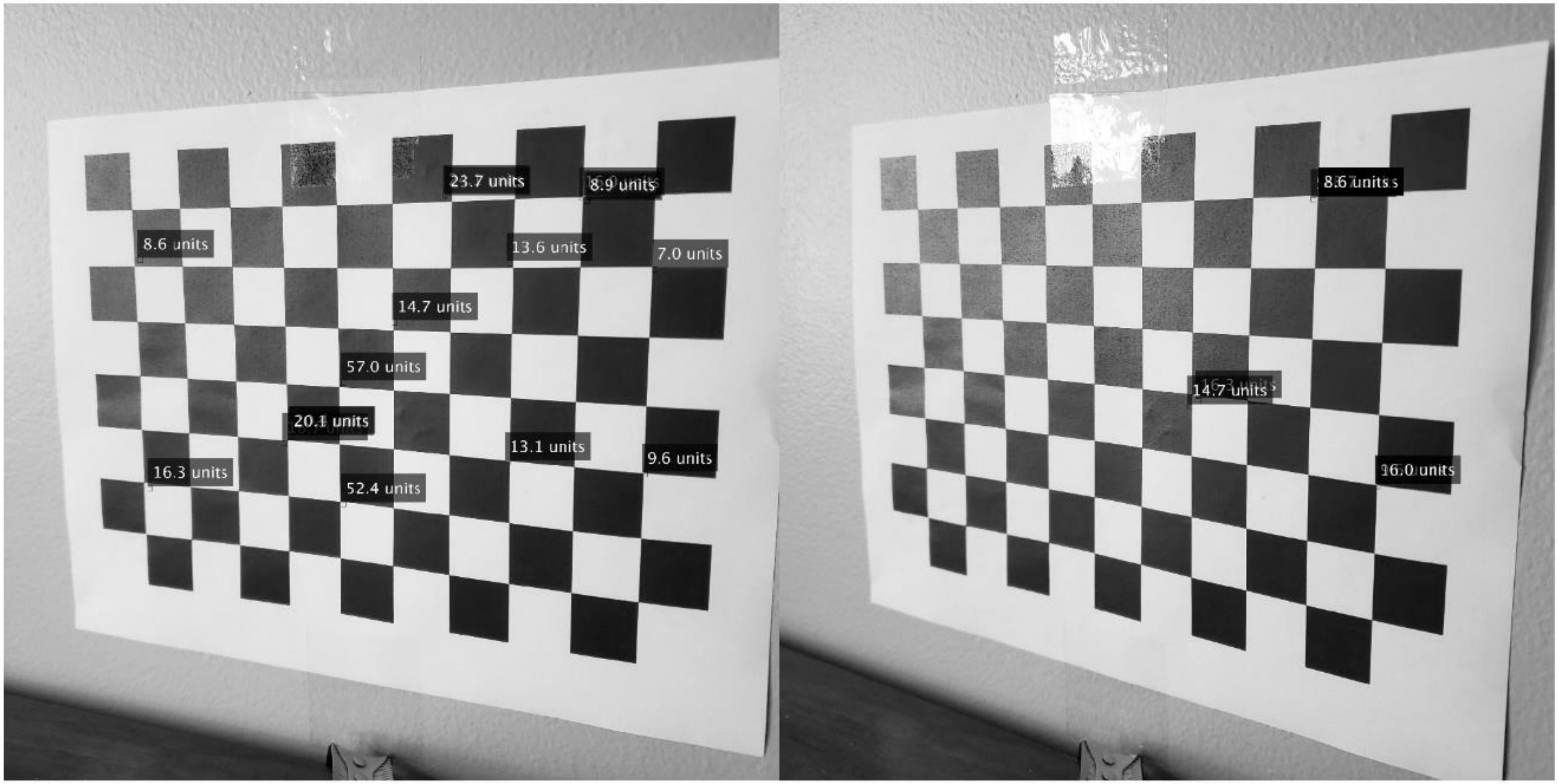

Figure 8: For images 8 and 9, triangulated distances to the chessboard image. This was done with the application of the RANSAC method. See that nearly all values are near the 12-inches (units) sought. Note the spurious value of 57.0 in the left image remains.